TECHNICAL TIP 02
Calculated Columns

Summary

1. Introduction

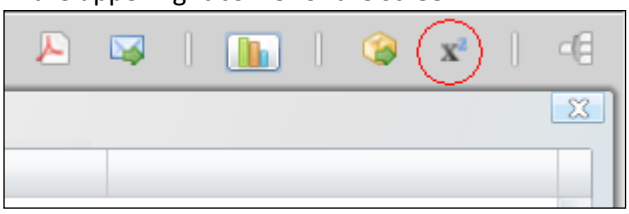
In BXBwebsuite the calculated columns are used to generate measures or dimensions based on formulas where the variables can be conditioned (if) or mathematical functions (Round that rounds a number, e.g.) and/or other measures or dimensions of the cube.

When a calculated column is created, it turns part of the cube, as a measure or dimension, like any other column.

2. Creating a calculated column

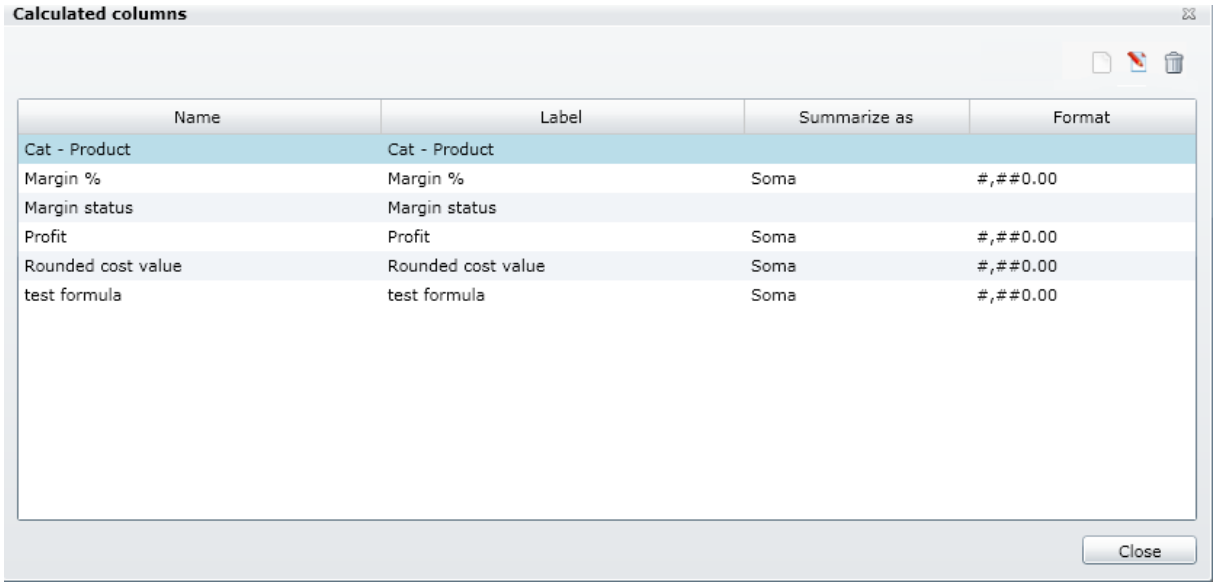
A calculated column can be created in 2 different points:

a) In *Reports* module of BXBreports, where you just need to open a report and click on the icon *Calculated columns*, in the upper right corner of the screen.

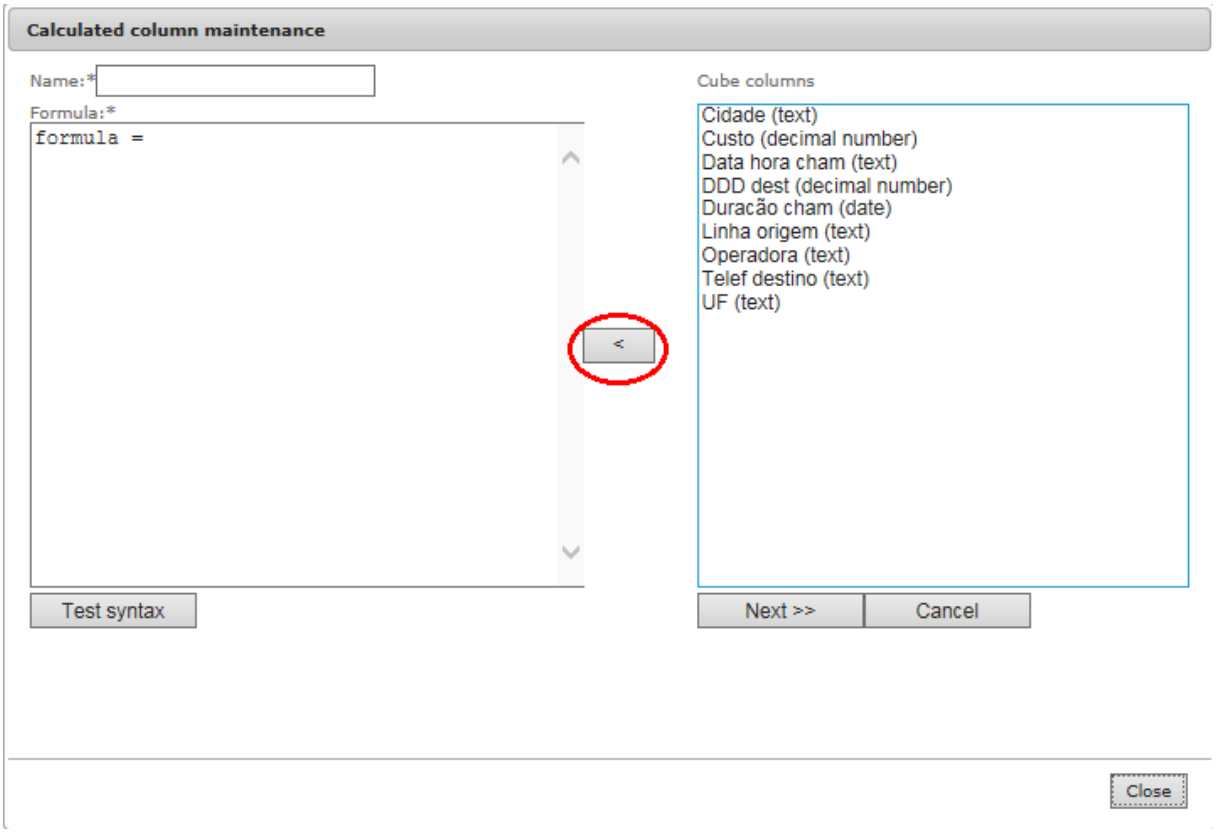


b) In *Database* module where, positioned in the cube configuration screen, the same icon appears in the upper right corner of the screen.

In *Calculated columns* screen, the list of calculated columns to the respective cube is shown and also the buttons to include, change or delete a calculated column. In this example, we are going to include a new calculated column by clicking the button "+", in the upper right corner of the screen.



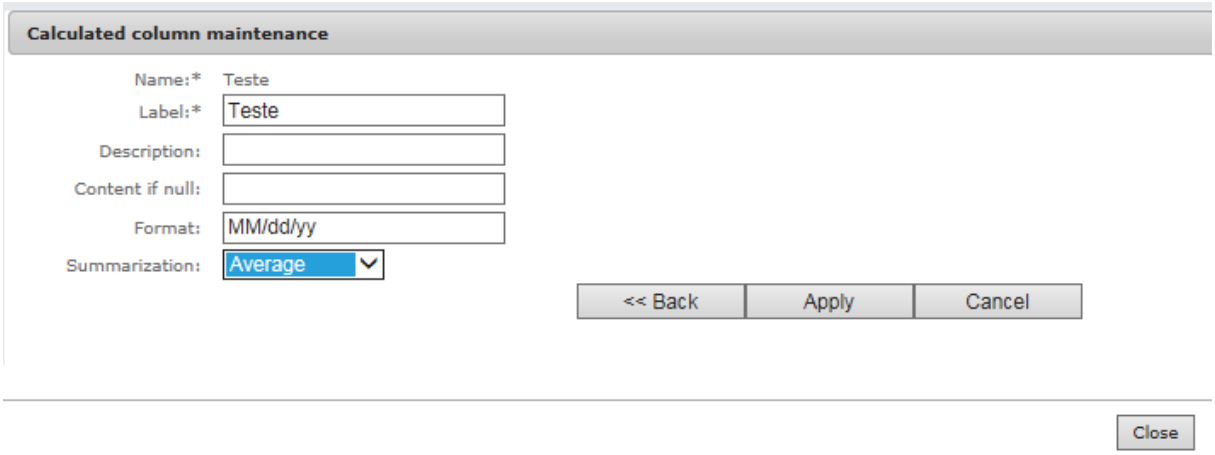
In the first part of the *Calculated column* maintenance screen there are fields to manage the column's name and the formula. On the right, you see a list of the cube's columns (measures and dimensions) that can be used in the new calculated column formulation (note that other calculated columns also appear in this listing because they are measures or dimensions too).
To add them to the expression, just select the chosen item and click the button *Use column* (<). To verify if the syntax is correct, just click on *Test syntax*.



Note: the respective data type is exhibited near each column to assist the formula construction.

In the second part (exhibited by clicking the button Next >>), you find the other calculated columns properties, as:
Label: may be filled with alternative name, being displayed as the title of the report column.
Description: optional fill (for documentation).
Content if null: alternative value, if the calculation returns "NULL".
Format: examples of completing the format field, according to the data type:
-Integers: ###.##0
-Decimals: ###.##0,00
Note: in the report, # will be replaced by white space.
-Date and time: MM/dd/yy HH:mm:SS
Note: see that MM represents month and mm represents minute.
Default summarization:
-Sum: displays the accumulated values in the reports where this column is totaled.

-Average: displays the average value in the reports where this column is totaled.
-Formula: displays the result of the calculation in the reports where this column is totaled.



3. Syntax

To create a calculated column the syntax used is ***result = expression;***. The value of the calculated column will be the value assigned to the variable *result,* in the introduced expression. The calculated column expression can be composed of operations among measures, dimensions and fixed values. This operation involves arithmetical, relational and logical operators.

Example (MARGIN%):
*result = (get('Sales Value) - get('Cost Value')) / get('Sales Value') * 100;*

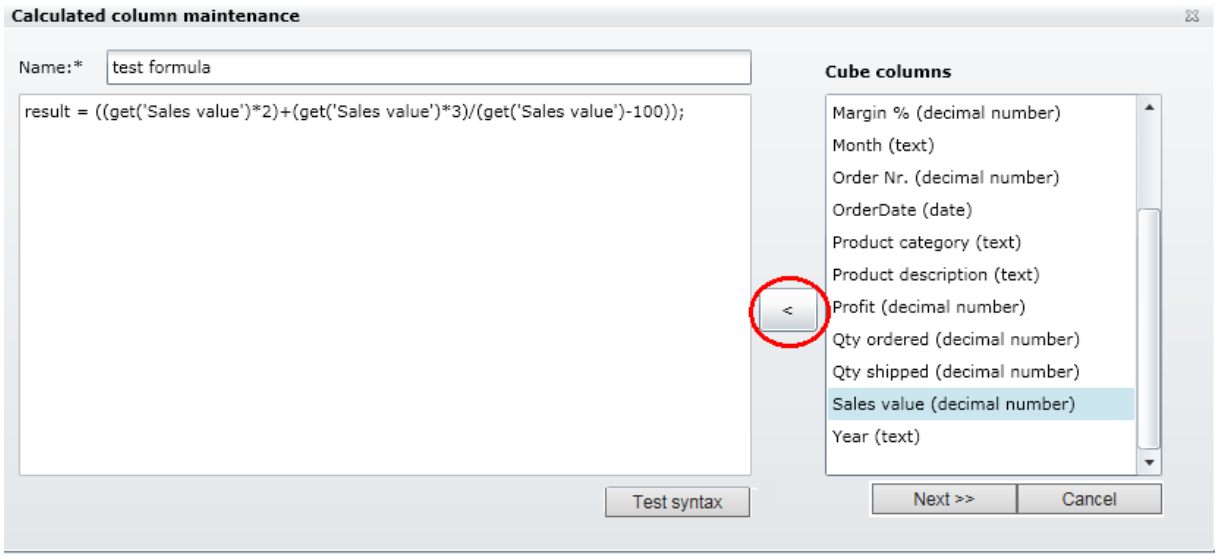
Tip: it is not necessary to type the calculated columns name according to the syntax above. You only have to choose the desired column and press the button Use column (<) in the Calculated column maintenance screen and the column name will be insert as the required syntax.

4. Common applications of calculated columns

4.1 Using arithmetic operators

Exemplify the use of arithmetic operators (multiplication, division, addition, subtraction) applying operator precedence by using parentheses.

*result = ((get('Sales value')*2)+(get('Sales value')*3)/(get('Sales value')-100));*



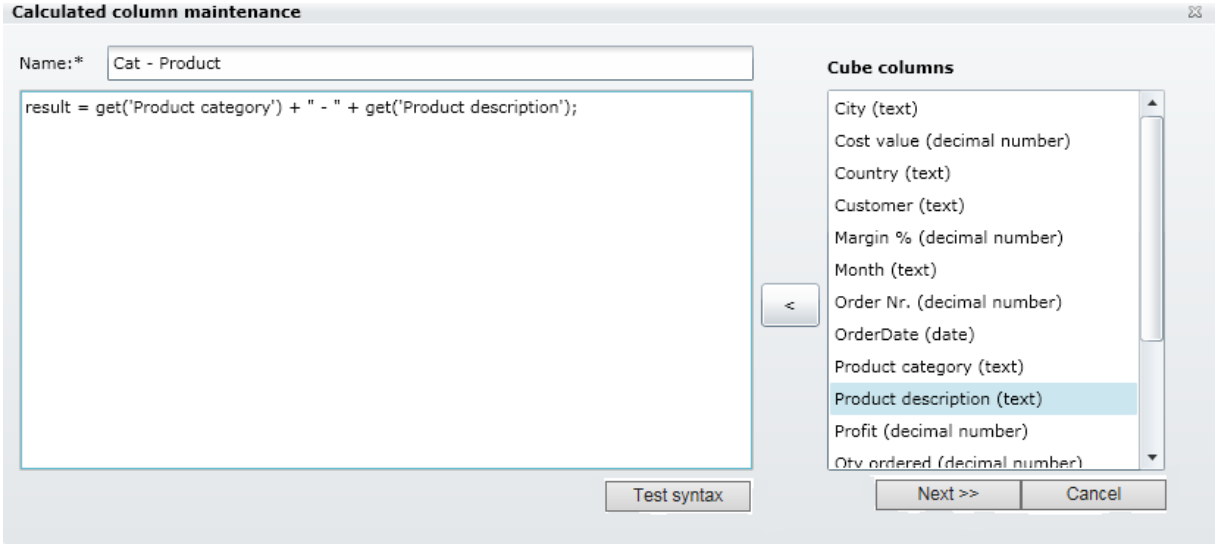
Result:

Product description	Sales value	test formula
5-SHELF WHITE MEMAMINE BOOKCASE	5.841,00	11.685,05
ADIRONDACK CHAIR	21.255,00	42.513,01
ADIRONDACK FOOTREST	255,00	514,94
ADIRONDACK TABLE	21.715,00	43.433,01
ANTIGUA NESTING TABLES (SET)	396,00	796,01
ANTIGUA SETTEE (BROWN)	3.781,00	7.565,08
BENTWOLD SETTEE	9.576,00	19.155,03
BISTRO DINING SET FOR TWO	42.042,00	84.087,01
BLACK HALOGEN FREE-STANDING LAMP	2.759,00	5.521,11
CERAMIC LAMP (ALL COLORS)	783,00	1.569,44
CHERRY COFFEE TABLE	500,00	1.003,75
CHERRY END TABLE	34.515,00	69.033,01

4.2 Concatenating text

Exemplify the dimensions *Product category* and *Product description* concatenation, separated by a hyphen.

```
result = get('Product category') + " - " + get('Product description');
```



Result:

Product category	Product description	Cat - Product
DESKS	2-Drawer Danish Desk	DESKS - 2-Drawer Danish Desk
SOFAS	3-Cushion Blue Fabric Sofa	SOFAS - 3-Cushion Blue Fabric Sofa
DESKS	3-Drawer Student Desk with Hutch	DESKS - 3-Drawer Student Desk with Hutch
DESKS	3-Piece Mobile Student's Desk	DESKS - 3-Piece Mobile Student's Desk
MISC	5-Shelf White Memamine Bookcase	MISC - 5-Shelf White Memamine Bookcase
CHAIRS	Adirondack Chair	CHAIRS - Adirondack Chair
MISC	Adirondack Footrest	MISC - Adirondack Footrest
TABLES	Adirondack Table	TABLES - Adirondack Table
TABLES	Antigua Nesting Tables (Set)	TABLES - Antigua Nesting Tables (Set)
SOFAS	Antigua Settee (Brown)	SOFAS - Antigua Settee (Brown)
SOFAS	Bentwold Settee	SOFAS - Bentwold Settee
TABLES	Bistro Dining Set for Two	TABLES - Bistro Dining Set for Two
LAMPS	Black Halogen Free-standing Lamp	LAMPS - Black Halogen Free-standing Lamp
LAMPS	Ceramic Lamp (All Colors)	LAMPS - Ceramic Lamp (All Colors)
TABLES	Cherry Coffee Table	TABLES - Cherry Coffee Table
TABLES	Cherry End Table	TABLES - Cherry End Table

ATTENTION: use quotation marks as a delimiter of fixed text to be concatenated, as in the example above, where the hyphen is between quotation marks. If the values to

be concatenated are measures and you use apostrophes as delimiter, the result will be the sum of the values with the ASCII value of the character (in this case, the hyphen, which has value 45).

4.3 Substring

To extract part of a text, use the *Substring* function, with the following syntax:

formula = (get('year-month')).Substring(startIndex, length);

startIndex: initial position, where the first position is 0
length: number of characters to be extracted (resulting field size)

For example, if you have a "year-month" text column with the content in "2018-08" format, to extract the month you must do:

formula = (get('year-month')).Substring(5,2);

4.4 Converting data types

To convert a numeric data to text, just use the expression *Convert.ToString(value)*. To convert from text to numeric, we can use:

Type	**Possible values to store**	**Command**
Int	-2,147,483,648 a 2,147,483,647 (32 bits)	*Convert.ToInt32(*value*)*
Float	±1.5 × 10−45 a ±3.4 × 1038 (32 bits)	Convert.ToSingle(value)
Double	±5.0 × 10−324 a ±1.7 × 10308 (64 bits)	*Convert.ToDouble(*value*)*
Decimal	±1.0 × 10−28 a ±7.9 × 1028 (128 bits)	*Convert.ToDecimal(*value*)*

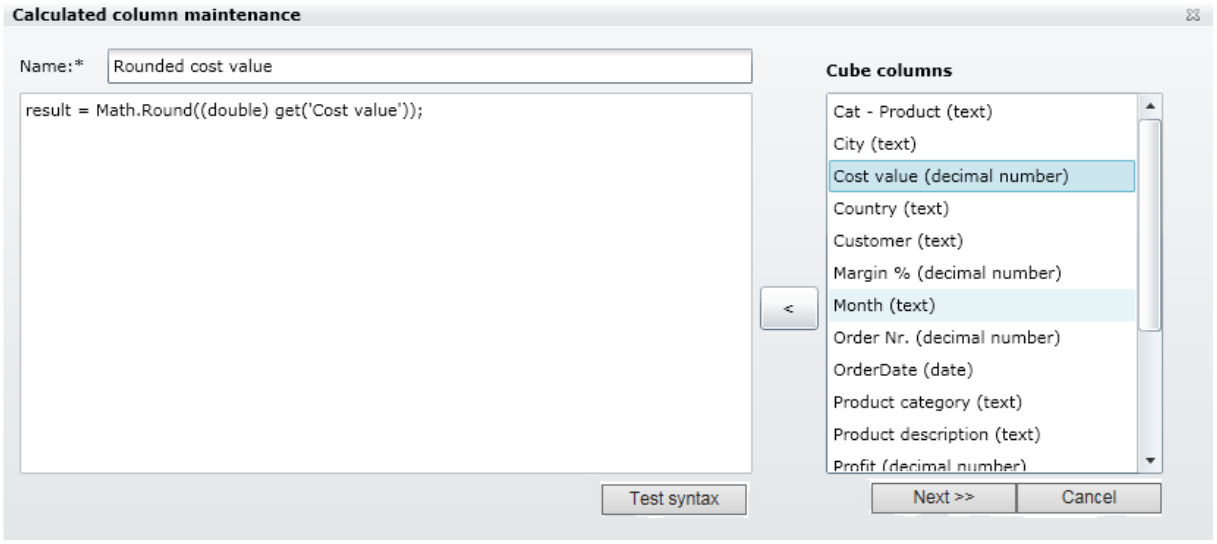
Ex. *result = get('PRODUCT') + Convert.ToString(get('Profit'));*

4.5 Using ready mathematical functions from Math class

When you create a calculated column in BXBSuite, it is possible to use Math class, composed by constant and functions that turn easier many mathematical operations.

In the example below, the function Round is used to round the value of the measure *Cost Value*. Note that in this calculated column there is a conversion of data types. For further information, see item 5a of the FAQ section. To see the full list of constants and functions of Math class, see item ***Appendix 1 - Math class members***.

result = Math.Round((double) get('Cost value'));



Result:

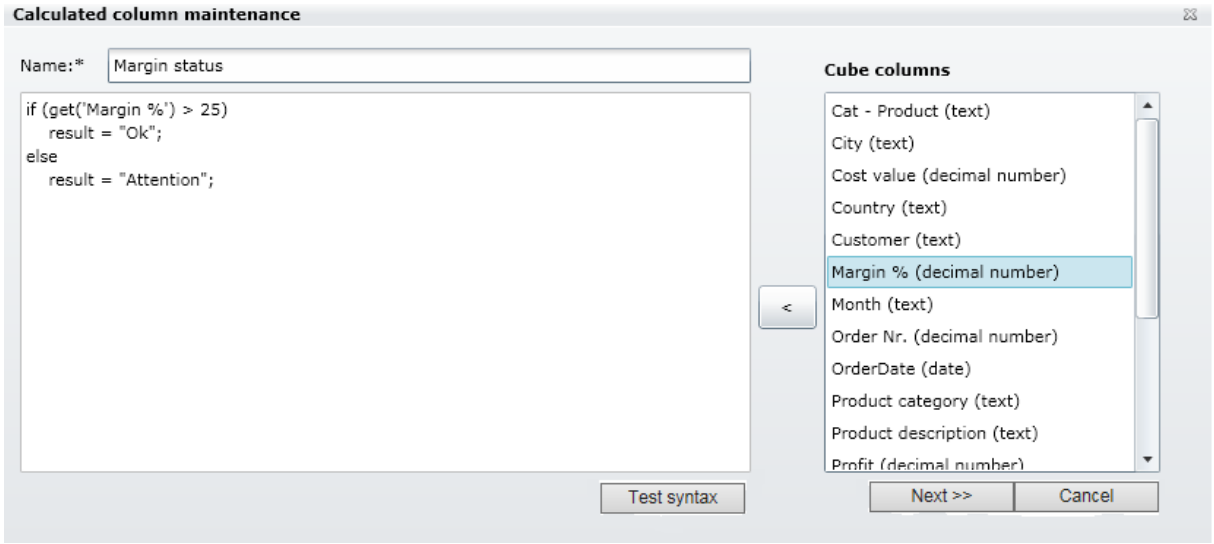
Product category	Product description	Cost value	Rounded cost value
SOFAS	3-Cushion Blue Fabric Sofa	30.380,90	30.381,00
SOFAS	Antigua Settee (Brown)	2.508,00	2.508,00
SOFAS	Bentwold Settee	6.564,90	6.565,00
SOFAS	Courting Swing	63.545,40	63.545,00
SOFAS	Devonaise Chaise	10.759,80	10.760,00
SOFAS	Italian White Leather Sofa	45.824,10	45.824,00
SOFAS	Pillow-Back Chaise (Paisley Pattern)	104.802,50	104.802,00
SOFAS	Pillow-Back Sofa (Paisley Pattern)	171.340,40	171.340,00
SOFAS	Water Hyacinth Loveseat	42.453,60	42.454,00
TOTAL		478.179,60	478.179,00

4.6 Using if-else

Beyond the arithmetic, relational and logical operators, it is also possible to use measures/dimensions to apply conditions that will influence the result, using the *if else* syntax.

In the example below, if the measure value MARGIN% is greater than 25, the result takes the value "OK", else, it takes the value "Attention."

```
if (get('Margin %') > 25)
   result = "Ok";
else
   result = "Attention";
```



This calculated column could also be written using ternary operator (compact way to express if-else), as follows:

result = get('Margin %') > 25 ? "Ok" : " Attention ";

Country / City	Margin %	Margin status
USA	21,49	Attention
ALBUQUERQUE	22,15	Attention
ALOHA	21,82	Attention
ANACORTES	21,82	Attention
ANCHORAGE	29,93	Ok
AUBURN	22,92	Attention
AUSTIN	22,15	Attention
BELLEVUE	21,82	Attention
BELLINGHAM	15,25	Attention
BEND	21,82	Attention
BERKELEY	22,15	Attention
BOISE	15,25	Attention
BOSTON	15,25	Attention
BOTHELL	15,40	Attention
BUFFALO	22,15	Attention
BUTTE	21,82	Attention
CHELAN	29,84	Ok
CHICAGO	21,82	Attention
DENVER	15,40	Attention
ELGIN	22,92	Attention
EUGENE	21,82	Attention

A dimension-condition example:

formula= get('status') == "Overdue" ? get('Invoice_amount') : 0

4.7 Compound formulas with AND / OR

"AND" is represented by && (a double ampersand sign)
"OR" is represented by ||

Examples for calculation an overdue value:

formula= get('Due_date') < DateTime.Today && (get('receipt_date') == Convert.ToDateTime("1900-01-01")) ? get('Invoice_amount') : 0

formula= get('Due_date') > DateTime.Today || (get('receipt_date') <> Convert.ToDateTime("1900-01-01")) ? 0 : get('Invoice_amount')

4.8 Entering constants with decimal places

Example: to get 10% of the *Sales Value*

*result = get (' Sales Value ') * (decimal) 0.1;*

4.9 Calculating in days the difference between 2 dates

result = (get('Final_date') - get('Initial_date')).Days;

4.10 Add/subtract n days from a date

result = DateTime.Today.AddDays(-30);
result = (get('SalesMonth ')).AddDays(30);

4.11 Identify previous month

result = ((get('SalesMonth ')).Year == DateTime.Today.AddMonths(-1).Year &&
(get('SalesMonth ')).Month == DateTime.Today.AddMonths(-1).Month) ? 'Last month ' :
'Other month ';

4.12 Convert time with HH:MM or HH:MM:SS format in seconds

formula = TimeToSeconds(get('timeColumn'))

4.13 Convert time with HH:MM or HH:MM:SS format in minutes

formula = TimeToMinutes(get('timeColumn'))

4.14 Convert time with HH:MM or HH:MM:SS format in hours

formula = TimeToHours(get('timeColumn'))

5. FAQ

a. Add a calculated column displays an error like "Best overloaded method compatible with 'System.Math.Sqrt (double)' has some invalid arguments. Argument '1 ': cannot convert from' decimal 'to' double '.". What should I do?

Answer: This error occurs because the *Sqrt* method of the Math class expects a numeric value of type *double* as a parameter, and the parameter used is a numeric of type *decimal*. To run the calculated column correctly, just do a typecast (data type conversion), as follows: "System.Math.Sqrt((double)*parameter*)".

Appendix 1 - Math class members

Function	Description	Syntax
Abs	Returns the absolute value of a specified number.	*Math.Abs(x)* *Where x is numeric, of type double.*
Acos	Returns the angle whose cosine is the specified number.	*Math.Acos(x)* *Where x is numeric, of type double.*
Asin	Returns the angle whose sine is the specified number.	*Math.Asin(x)* *Where x is numeric, of type double.*
Atan	Returns the angle whose tangent is the specified number.	*Math.Atan(x)* *Where x is numeric, of type double.*
Atan2	Returns the angle whose tangent is the quotient of two specified numbers.	*Math.Atan2(x, y)* *Where x and y are numeric, of type double.*
BigMul	Produces the full product of two 32-bit numbers.	*Math.BigMul(x, y)* *Where x and y are numeric, of type int.*
Ceiling	Returns the smallest integer greater than or equal to the specified number.	*Math.Ceiling(x, y)* *Where x and y are numeric, of type double or decimal.*
Cos	Returns the cosine of the specified angle.	*Math.Cos(x)* *Where x is numeric, of type double.*
Cosh	Returns the hyperbolic cosine of the specified angle.	*Math.Cosh(x)* *Where x is numeric, of type double.*
DivRem	Calculates the quotient of two numbers and also returns the remainder in an output parameter.	*Math.DivRem(x, y, z)* *Where x, y and z are numeric, of type int.*

Exp	Returns *e* raised to the specified power.	*Math.Exp(x)* *Where x is numeric, of type double.*
Floor	Returns the largest integer smaller than or equal to the specified number.	*Where x and y are numeric, of type double or decimal.*
IEEERemainder	Returns the specified number of the remainder resulting from the division of a specified number by another.	*Math.IEEERemainder (x, y)* *Where x and y are numeric, of type double.*
Log	Returns the logarithm of a specified number in a specified base.	*Math.Log(x) ou Math.Log(x, y)* *Where x and y are numeric, of type double.*
Log10	Returns the base 10 logarithm of a specified number.	*Math.Log10(x)* *Where x is numeric, of type double.*
Max	Returns the larger of two numbers.	*Math.Max (x, y)* *Where x and y are numeric, of type int, double or decimal.*
Min	Returns the smaller of two numbers.	*Math.Min (x, y)* *Where x and y are numeric, of type int, double or decimal.*
Pow	Retorna um número especificado elevado à potência especificada.	*Math.Powx (x, y)* *Where x and y are numeric, of type double.*
Round	Rounds a value to the nearest integer or specified number of decimal places.	*Math.Round (x) or Math.Round (x, y)* *Where x and y are numeric, x of type double or decimal and y of type int.*
Sign	Returns a value indicating the sign of a number.	*Math.Sign (x)* *Where x is numeric, of type int, double or decimal.*
Sin	Returns the sine of the	*Math.Sin(x)*

	specified angle.	*Where x is numeric, of type double.*
Sinh	Returns the hyperbolic sine of the specified angle.	*Math.Sinh(x)* *Where x is numeric, of type double.*
Sqrt	Returns the square root of the specified number.	*Math.Sqrt(x)* *Where x is numeric, of type double.*
Tan	Returns the tangent of the specified angle.	*Math.Tan(x)* *Where x is numeric, of type double.*
Tanh	Returns the hyperbolic tangent of the specified angle.	*Math.Tanh(x)* *Where x is numeric, of type double.*
Truncate	Calculates the integral part of a number.	*Math.Truncate(x)* *Where x is numeric, of type double or decimal.*

C

Constant	Description	Value	Syntax
E	Represents the natural logarithmic base, especified by the constant *e*.	2.7182818284590452354	Math.E
PI	Represents the proportion between the circumference of a circle and its diameter, specified by the constant *π*.	3.14159265358979323846	Math.PI