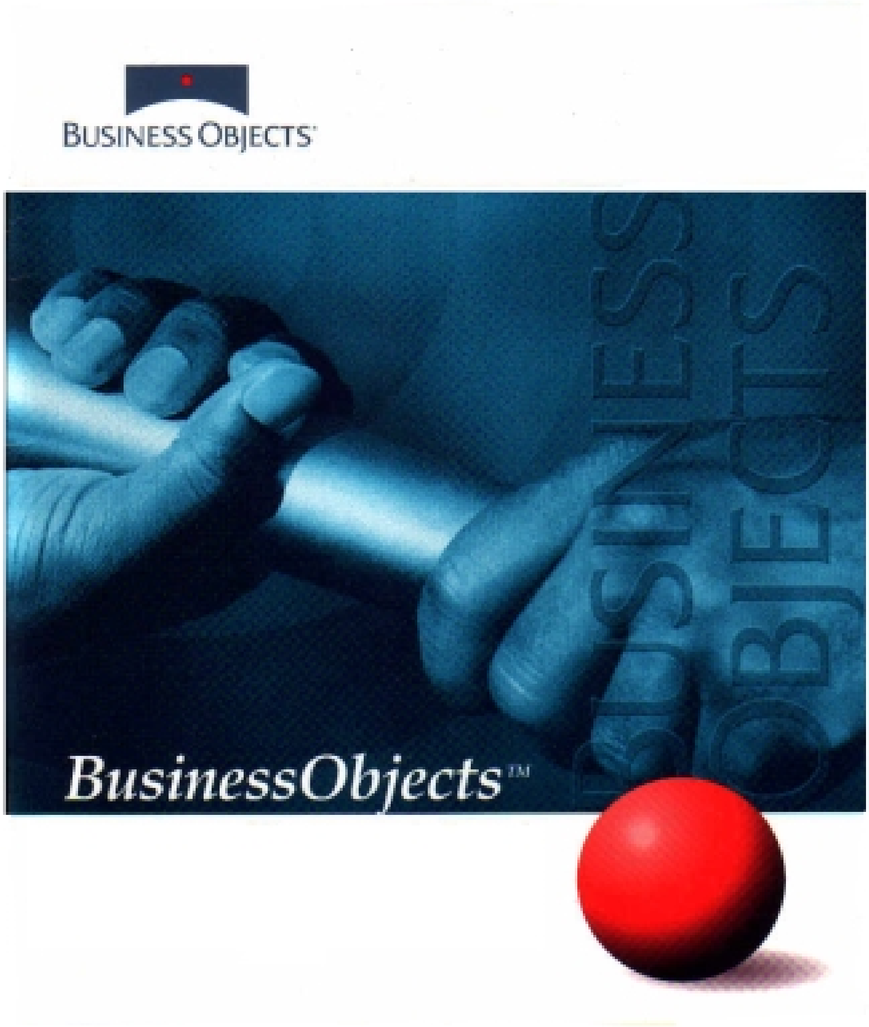


Guide to Formulas in the Edit Formula in the Business Object Reporter Module

Business Objects Reporter
4.1.x & 5.x

Originally created on: 30/11/98

Updated on: 20/9/2000 in order to take into account new V5.x Functions

Created by: Suzanna Rahimi

Business Objects
4.1.x & 5.x

Guide to Formulas in the Edit Formula in the Business Object Reporter Module

Permission to use this document is authorized, provided that A) the use of the document is for informational and non-commercial purposes, and B) the document retain the copyright notice contained in this disclaimer. Business Objects may modify this document at any time and without notice. THIS DOCUMENT IS DELIVERED "AS IS" AND WITHOUT WARRANTY OF ANY KIND INCLUDING, BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL BUSINESS OBJECTS, ITS SUPPLIERS OR OTHER THIRD PARTIES MENTIONED HEREIN BE LIABLE FOR ANY DAMAGES ARISING FROM THE USE OF THIS DOCUMENT INCLUDING, BUT NOT LIMITED TO, SPECIAL, INDIRECT, PUNATIVE OR CONSEQUENTIAL DAMAGES. The documentation, and any software to which it may apply, shall be deemed "commercial computer software" and "commercial computer software documentation" pursuant to DFAR Section 227.7202 and FAR Section 12.212 (and the successor to such sections, if any), as applicable. The use of any documentation including, but not limited to, its reproduction and display, by the United States of America, its departments, agencies and/or any of its instrumentalities, regardless of form, shall be governed as set forth herein.

Table of Contents

Introduction **6**

1. Aggregates **7**

1.1 Average 8
1.2 Count 10
1.3 Max 14
1.4 Min 16
1.5 RunningAverage 18
1.6 RunningCount 21
1.7 RunningMax 23
1.8 RunningMin 24
1.9 RunningSum 25
1.10 StdDev 29
1.11 StdDevP 34
1.12 Sum 39
1.13 Var 41
1.14 VarP 43

2 Numeric functions **45**

2.1 Abs 46
2.2 Ceil 47
2.3 Cos 48
2.4 Exp 49
2.5 Fact 50
2.6 Floor 51
2.7 Ln 52
2.8 Log 53
2.9 Log10 54
2.10 Mod 55
2.11 Power 56
2.12 Rank 57
2.13 Round 58
2.14 ServerValue 59
2.15 Sign 60
2.16 Sin 61
2.19 Sqrt 62
2.20 Tan 63
2.21 ToNumber 64
2.22 Truncate 65

3 Character Functions **66**

3.1 Asc 67
3.2 Char 68
3.3 Concatenation 69
3.4 Fill 71
3.5 FormatDate 72
3.6 FormatNumber 74
3.7 InitCap 75
3.8 Left 76
3.9 LeftPad 77

3.10 LeftTrim ... 78
3.11 Length ... 79
3.12 Lower ... 80
3.13 Match ... 81
3.14 Pos ... 82
3.15 Replace ... 83
3.16 Right ... 84
3.17 RightPad ... 85
3.18 RightTrim ... 86
3.19 SubStr ... 87
3.20 Trim ... 88
3.21 Upper ... 89
3.22 WordCap ... 90

4 Date Functions ... 91

4.1 CurrentDate ... 92
4.2 CurrentTime ... 93
4.3 DayName ... 94
4.4 DayNumberofMonth ... 95
4.5 DayNumberOfWeek ... 96
4.6 DayNumberOfYear ... 97
4.7 DaysBetween ... 98
4.8 LastDayOfMonth ... 99
4.9 LastDayOfWeek ... 100
4.10 Month ... 101
4.11 MonthNumberOfYear ... 102
4.12 MonthsBetween ... 103
4.13 Quarter ... 104
4.14 RelativeDate ... 105
4.15 ToDate ... 106
4.16 Week ... 108
4.17 Year ... 109

5 Logical Functions ... 110

5.1 IsDate ... 111
5.2 IsError ... 112
5.3 IsLogical ... 113
5.4 IsNumber ... 114
5.5 IsNull ... 115
5.6 IsString ... 116
5.7 IsTime ... 117

6 Document Functions ... 118

6.1 BlockNumber ... 119
6.2 ColumnNumber ... 120
6.3 DocumentAuthor ... 121
6.4 DocumentDate ... 122
6.5 DocumentName ... 123
6.6 DocumentPartiallyRefreshed ... 124
6.7 DocumentTime ... 125
6.8 LastPrintDate ... 126
6.9 LineNumber ... 127
6.10 NumberOfPages ... 128
6.11 Page ... 129
6.12 SectionNumber ... 130

7 Data Provider Functions 131

7.1 Connection 132
7.2 DataProvider 133
7.3 DataProviderSQL 134
7.4 DataProviderType 135
7.5 LastExecutionDate 136
7.6 LastExecutionTime 137
7.7 NumberOfDataProviders 138
7.8 NumberOfRows 139
7.9 UniverseName 140
7.10 UserResponse 141

8 Misc Functions 142

8.1 ApplicationValue 143
8.2 CurrentUser 145
8.3 GetProfileNumber 146
8.4 GetProfileString 147
8.5 NameOf 148
8.6 NoFilter 149
8.7 Previous 150
8.8 RowIndex 153

9 Additional Functions in V5.x 154

9.1 Aggregates 155
9.1.1 Count All 155
9.1.2 Product 156
9.1.3 RunningProduct 157

9.2 Numeric Functions 160
9.2.1 EuroConvertFrom 160
9.2.2 EuroConvertTo 162
9.2.3 EuroFromRoundErr 164
9.2.4 EuroToRoundErr 165
9.2.5 Median 166

9.3 Logical Functions 169
9.3.1 Even 169
9.3.2 Odd 170

9.4 Document Functions 171
9.4.1 DrillFilters 171
9.4.2 GlobalFilters 172
9.4.3 PageInSection 173

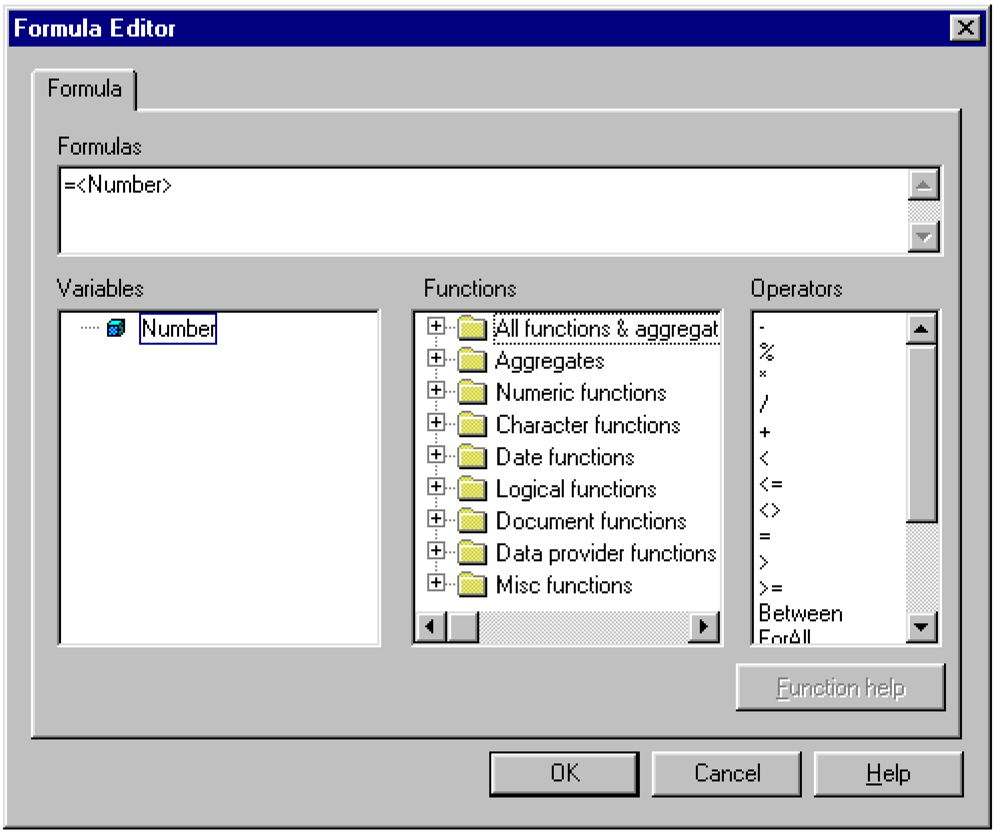
9.5 Misc Functions 174
9.5.1 MultiCube 174

Guide to Formulas in the Edit Formula in the Business Object Reporter Module

Introduction

The aim of this document is to provide further information on the use of the Formula Functions based in the Edit Formula Menu in the Reporter Module.
There are help files based on these Functions, but they do not always portray the Functions in full. In this guide I will try to provide adequate examples of the Formulas & Functions used in the Edit Formula.

This option can be found, when a right mouse click is applied after highlighting a column in a Business Objects report or in the Data Menu in the Taskbar.



Formulas: =<Number>

Variables: Number

This document will now concentrate on the individual formulas in the Edit Formula Menu.

1. Aggregates

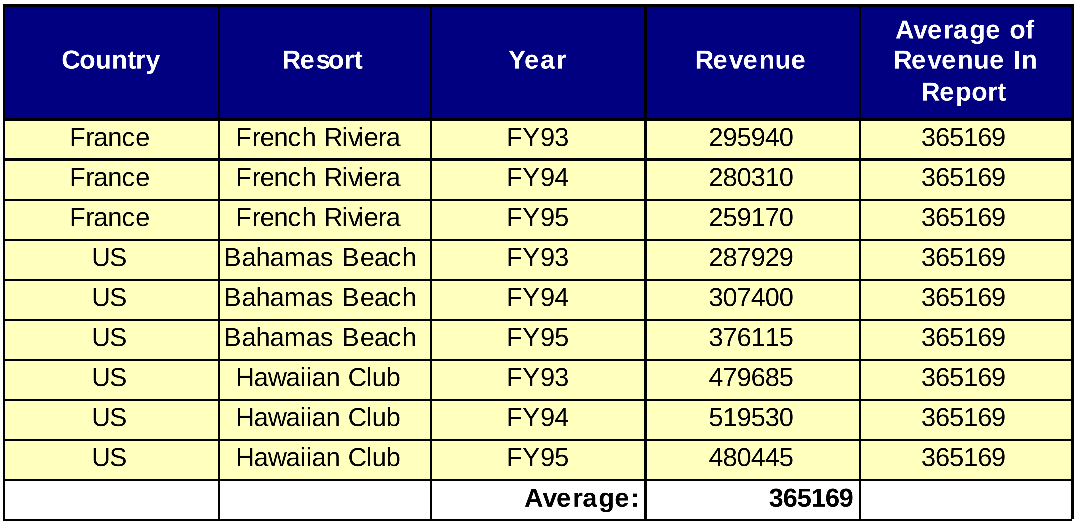
The Aggregates Functions are divided into the following Functions:

Average
Count
Max
Min
RunningAverage
RunningCount
RunningMax
RunningMin
RunningSum
StdDev
StdDevP
Sum
Var
VarP

This Chapter will now have a close look at the individual Aggregate Functions

1.1 Average

The average of an object can be obtained of any measure object or numeric value in the report. I have created an example here with the average of Revenue.
The average can be used in the same way as you would use the average function in Insert Calculations. You can insert a column in the report and request the:
=Average(<Revenue>) In Report.
In the report this will show the average as 365169.



Country	Resort	Year	Revenue	Average of Revenue In Report
France	French Riviera	FY93	295940	365169
France	French Riviera	FY94	280310	365169
France	French Riviera	FY95	259170	365169
US	Bahamas Beach	FY93	287929	365169
US	Bahamas Beach	FY94	307400	365169
US	Bahamas Beach	FY95	376115	365169
US	Hawaiian Club	FY93	479685	365169
US	Hawaiian Club	FY94	519530	365169
US	Hawaiian Club	FY95	480445	365169
		Average:	**365169**	

(Note: You would need the In Report as Business Objects calculates this per row. If it is excluded Business Objects will see all individual Revenue values as their own Average.)

The extended syntax for Aggregate Functions allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.
An example for this with the Average Function in this report will be to calculate the average per dimension, for example Country.

This is also the way to deal with averages per Break or per Section.

The Formula used for this would be:
=Average (<Revenue>) In <Country>
For France this would be:
(295940+280310+259170)/3=278473

Country	Resort	Year	Revenue	Average per Country	Average per Country and Resort
France	French Riviera	FY93	295940	278473	278473
	French Riviera	FY94	280310	278473	278473
	French Riviera	FY95	259170	278473	278473

Country	Resort	Year	Revenue	Average per Country	Average per Country and Resort
US	Bahamas Beach	FY93	287929	408517	323815
	Bahamas Beach	FY94	307400	408517	323815
	Bahamas Beach	FY95	376115	408517	323815
	Hawaiian Club	FY93	479685	408517	493220
	Hawaiian Club	FY94	519530	408517	493220
	Hawaiian Club	FY95	480445	408517	493220

Again this can be extended to the Average per Country and Resort, as illustrated in report.
The formula for this is:
=Average(<Revenue>) In (<Country>,<Resort>)
For Hawaiian Club this would be:
(479685+519530+480445)/3=493220
Any further (dimensional) extensions can be added, separated by a comma.

Examples:

=Average(<Revenue>) In (<Country>, <Year>)
=Average(<Revenue> In (<Country>,<Resort>)) In (<Year>)

1.2 Count

The Count Aggregate can be used to count number, character strings and date values.
The Count of Revenue will bring back the number of occurrences of Revenue in the report.

Country	Resort	Year	Revenue	Revenue Count
France	French Riviera	FY93	295940	1
France	French Riviera	FY94	280310	1
France	French Riviera	FY95	259170	1
US	Bahamas Beach	FY93	287929	1
US	Bahamas Beach	FY94	307400	1
US	Bahamas Beach	FY95	376115	1
US	Hawaiian Club	FY93	479685	1
US	Hawaiian Club	FY94	519530	1
US	Hawaiian Club	FY95	480445	1

As with all the aggregates, the extended syntax for Aggregate Functions allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

An example with the Count Function is, to calculate the count the number of Year occurrences per Country. The Formula used for this is:
=Count(<Year>) In <Country>

Country	Resort	Year	Revenue	Number of Year Occurances per Country
France	French Riviera	FY93	295940	3
France	French Riviera	FY94	280310	3
France	French Riviera	FY95	259170	3
US	Bahamas Beach	FY93	287929	3
US	Bahamas Beach	FY94	307400	3
US	Bahamas Beach	FY95	376115	3
US	Hawaiian Club	FY93	479685	3
US	Hawaiian Club	FY94	519530	3
US	Hawaiian Club	FY95	480445	3

Another example is to calculate the number of occurrences of the Year per Resort and per Country.
The formula will be as follows:
=Count(<Year>,<Resort>) In <Country>
The result is shown in the report illustration below.

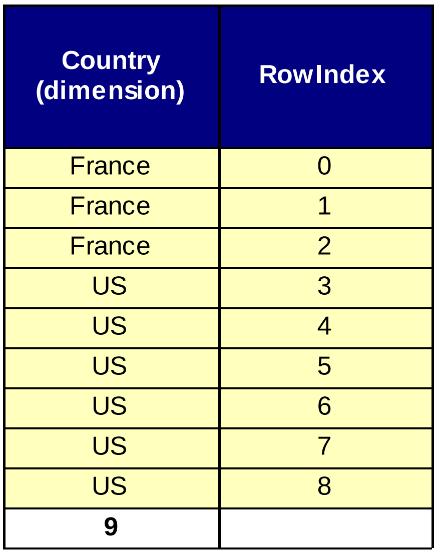
Country	Resort	Year	Revenue	Count of Year Occurances per Country and per Resort
France	French Riviera	FY93	295940	3
France	French Riviera	FY94	280310	3
France	French Riviera	FY95	259170	3
US	Bahamas Beach	FY93	287929	6
US	Bahamas Beach	FY94	307400	6
US	Bahamas Beach	FY95	376115	6
US	Hawaiian Club	FY93	479685	6
US	Hawaiian Club	FY94	519530	6
US	Hawaiian Club	FY95	480445	6

Note: If the count function of the Insert Calculations is used, this will display the Count Distinct values when applied on a dimension (numeric, character string or as a date).
For example a count on the Country column will give a count of 2, as there are 2 distinct values. When applied on a measure however it will give a count of all the rows even when the data is identical in some rows.

Country (dimension)	Revenue (measure)
France	295940
France	280310
France	259170
US	287929
US	479685
US	307400
US	519530
US	376115
US	480445
2	**9**

A work around for this is to create a column with a unique value variable (for example Rowindex set as a variable) and set the formula as follows:

=Count(<Resort>,<Rowindex>)



Country (dimension)	RowIndex
France	0
France	1
France	2
US	3
US	4
US	5
US	6
US	7
US	8
9	

This will return the count result as 9.

Alternatively a double count also brings back the entire row count: =Count(Count(<Resort>))

In V5 there is a new Function for this called CountAll()
(See chapter on Additional V5 Functions)

There is one other difference with the Count Function in V4 & V5.
In V4 when a count of occurrences is shown in a report, based on a numeric dimension that contains #empty values it will still count the #empty as an occurrence, marking this by the count of 1 In V5 the #empty are not taken as occurrence and are counted as 0:

result in v4:

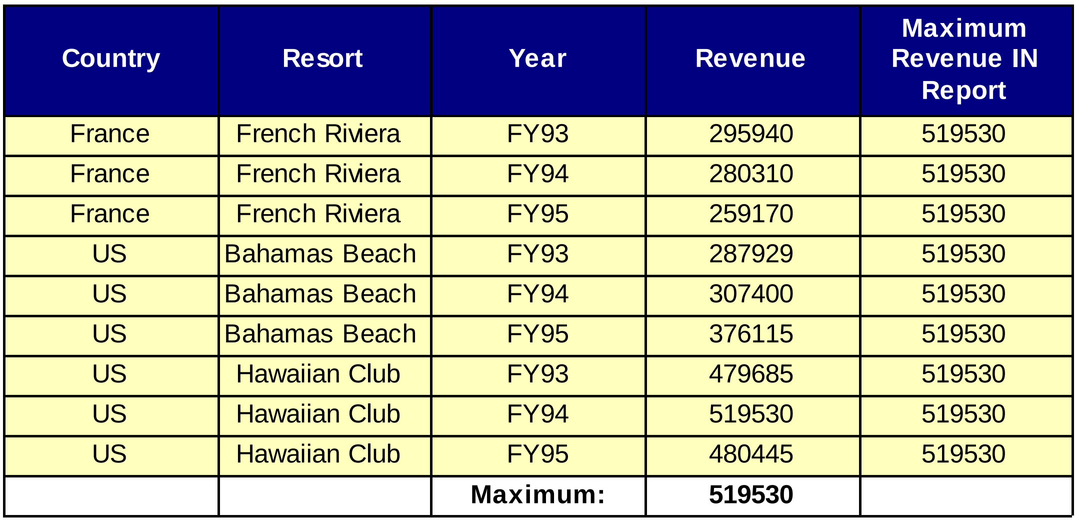
Country	Null Test	Count of Null Test
France	FY94	1
France	FY95	1
France		1
US	FY94	1
US	FY95	1
US		1

result in v5:

Country	Null Test	Count of Null Test
France	FY94	1
France	FY95	1
France		0
US	FY94	1
US	FY95	1
US		0

1.3 Max

This Aggregate Function stands for the calculation of the maximum value of a dimensional number value or a measure value.
As with the Average Function, the Max Function can be used in the same way as you would use the Max Function in Insert Calculations.
You can insert a column in the report and request the:
=Max(<Revenue>) In Report.
In the report this will show the maximum as 519530.



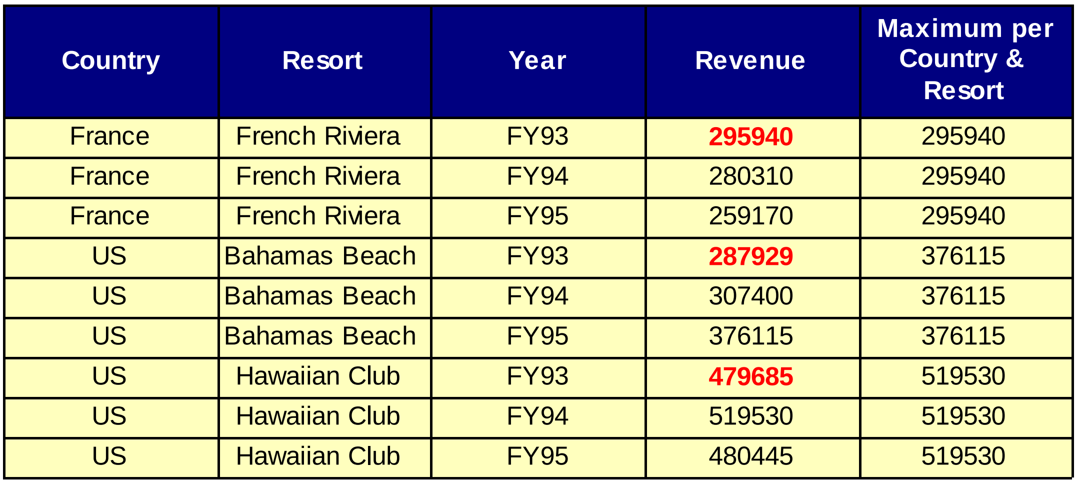
Country	Resort	Year	Revenue	Maximum Revenue IN Report
France	French Riviera	FY93	295940	519530
France	French Riviera	FY94	280310	519530
France	French Riviera	FY95	259170	519530
US	Bahamas Beach	FY93	287929	519530
US	Bahamas Beach	FY94	307400	519530
US	Bahamas Beach	FY95	376115	519530
US	Hawaiian Club	FY93	479685	519530
US	Hawaiian Club	FY94	519530	519530
US	Hawaiian Club	FY95	480445	519530
		Maximum:	**519530**	

(Note: You would need the In Report as Business Objects calculates this per row. If it is excluded Business Objects will see all individual Revenue values as their own Maximum.)

This Aggregate Function, as with all other Aggregates can be extended to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

An example of this is to calculate the:

=Max(<Revenue> In (<Country,<Resort>)
This example computes a table showing the maximum revenue in each of the Countries for the Resort in the Country with the most revenue. See results in the report below:



Country	Resort	Year	Revenue	Maximum per Country & Resort
France	French Riviera	FY93	**295940**	295940
France	French Riviera	FY94	280310	295940
France	French Riviera	FY95	259170	295940
US	Bahamas Beach	FY93	**287929**	376115
US	Bahamas Beach	FY94	307400	376115
US	Bahamas Beach	FY95	376115	376115
US	Hawaiian Club	FY93	**479685**	519530
US	Hawaiian Club	FY94	519530	519530
US	Hawaiian Club	FY95	480445	519530

For French Riviera the maximum will be 295940, for Bahamas Beach 376115, and for Hawaiian Club this is 519530.
The same calculation can be done with the Min Function in order to return the minimum revenue.

1.4 Min

This Aggregate Function stands for the calculation of the minimum value of a dimensional number value or a measure value.
As with the Max Function the Min Function can be used in the same way as you would use the Minimum Function in Insert Calculations.
You can insert a column in the report and request the:
=Min(<Revenue>) In Report.
In the report this will show the minimum as 259170.

Country	Resort	Year	Revenue	Minimum Revenue IN Report
France	French Riviera	FY93	295940	259170
France	French Riviera	FY94	280310	259170
France	French Riviera	FY95	259170	259170
US	Bahamas Beach	FY93	287929	259170
US	Bahamas Beach	FY94	307400	259170
US	Bahamas Beach	FY95	376115	259170
US	Hawaiian Club	FY93	479685	259170
US	Hawaiian Club	FY94	519530	259170
US	Hawaiian Club	FY95	480445	259170
		Minimum:	**259170**	

(Note: You would need the In Report as Business Objects calculates this per row. If it is excluded Business Objects will see all individual Revenue values as their own Minimum.)

The Min Function can, as with the other Aggregate Functions be used in an extended syntax, specified with In, ForEach, or ForAll operators.

For Example:

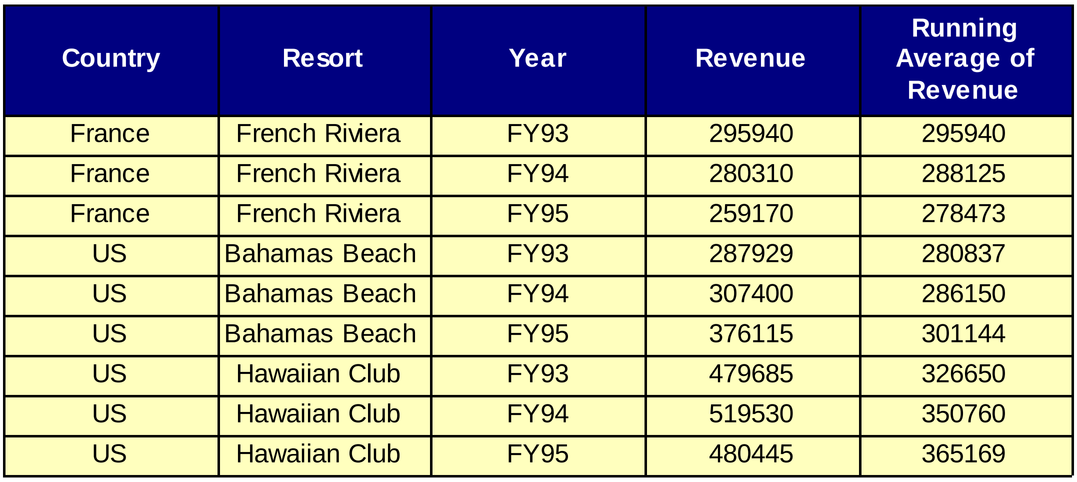
=Min(<Revenue>) ForAll (<Year>) In <Country>

Country	Resort	Year	Revenue	Minimum per Country & Resort
France	French Riviera	FY93	**295940**	259170
France	French Riviera	FY94	280310	259170
France	French Riviera	FY95	259170	259170
US	Bahamas Beach	FY93	**287929**	287929
US	Bahamas Beach	FY94	307400	287929
US	Bahamas Beach	FY95	376115	287929
US	Hawaiian Club	FY93	**479685**	479685
US	Hawaiian Club	FY94	519530	479685
US	Hawaiian Club	FY95	480445	479685

This will return the minimum revenue per year, which in this example is 1993(for both France and the US), per country. For France this is 259170, and for the US 287929.
The same calculation can be done with the Max Function, in order to return the maximum revenue, per year, per country.

1.5 RunningAverage

The RunningAverage will return a cumulative average of a numeric dimension or a measure.
The report below shows the RunningAverage of the Revenue



Country	Resort	Year	Revenue	Running Average of Revenue
France	French Riviera	FY93	295940	295940
France	French Riviera	FY94	280310	288125
France	French Riviera	FY95	259170	278473
US	Bahamas Beach	FY93	287929	280837
US	Bahamas Beach	FY94	307400	286150
US	Bahamas Beach	FY95	376115	301144
US	Hawaiian Club	FY93	479685	326650
US	Hawaiian Club	FY94	519530	350760
US	Hawaiian Club	FY95	480445	365169

The RunningAverage shows a cumulative average of the increase or decrease of the Revenue. The first Revenue value will be repeated in the RunningAverage column as it will be a calculation of: (295940+empty)/2 =295940. The second value will be (295940+280310)/2 = 288125.The third value will be (295940+280310+259170)/3 (Note that the end figure of the RunningAverage is the same figure as the Average figure out the Insert Calculations menu. This is correct as Business Objects will calculate the total sum of the revenue (3286524) divided by the total count of rows (9) which equals: 365169)

As with all the Aggregates, the formula can be extended with the use of the Context Operators ForEach, ForAll, and In.

If the formula is specified as:

=RunningAverage(<Revenue>) In <Year >

This will display the cumulative average of the average revenue per year. (For Y93 this is:(295940+287929+479685)/3=354518)

Country	Resort	Year	Revenue	Running Average per Year
France	French Riviera	FY93	295940	354518
France	French Riviera	FY94	280310	361799
France	French Riviera	FY95	259170	365169
US	Bahamas Beach	FY93	287929	354518
US	Bahamas Beach	FY94	307400	361799
US	Bahamas Beach	FY95	376115	365169
US	Hawaiian Club	FY93	479685	354518
US	Hawaiian Club	FY94	519530	361799
US	Hawaiian Club	FY95	480445	365169

If the calculation should show the RunningAverage reset per break or per section, in this case per Year, then the formula should be specified as:

=RunningAverage(<Revenue>;<Year>)

Year	Country	Resort	Revenue	Running Average Reset per Year
FY93	France	French Riviera	295940	295940
	US	Bahamas Beach	287929	291935
	US	Hawaiian Club	479685	354518
FY93				

Year	Country	Resort	Revenue	Running Average Reset per Year
FY94	France	French Riviera	280310	280310
	US	Bahamas Beach	307400	293855
	US	Hawaiian Club	519530	369080
FY94				

This applies to all Running aggregates, when a decrease or increase is to be calculated per section or break.
When resetting the calculation per row or column in a crosstab the semi-colon should also be used. The dimension behind the semi-colon will be the object that specifies where the reset should take place. In the example shown on the previous page Year is the dimension for which the reset is taking place. The calculation will therefore restart from each first value per Year.

1.6 RunningCount

The RunningCount will return a cumulative count of the number of occurrences of a numeric dimension or a measure.
When calculating the RunningCount of the Revenue it will return a run of occurrences, rather than the count of 1 for each occurrence as with the Count function (see chapter 1.2).

Country	Resort	Year	Revenue	Running Count
France	French Riviera	FY93	295940	1
France	French Riviera	FY94	280310	2
France	French Riviera	FY95	259170	3
US	Bahamas Beach	FY93	287929	4
US	Bahamas Beach	FY94	307400	5
US	Bahamas Beach	FY95	376115	6
US	Hawaiian Club	FY93	479685	7
US	Hawaiian Club	FY94	519530	8
US	Hawaiian Club	FY95	480445	9

The RunningCount Function can, as with the other Aggregate Functions be used in an extended syntax, specified with In, ForEach, or ForAll operators.

An example for this is:

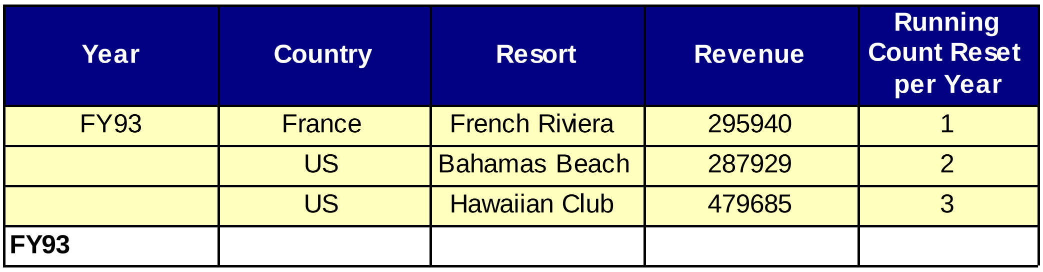
=RunningCount(<Revenue>) In <Year>

Country	Resort	Year	Revenue	Running Count per Year
France	French Riviera	FY93	295940	3
France	French Riviera	FY94	280310	6
France	French Riviera	FY95	259170	9
US	Bahamas Beach	FY93	287929	3
US	Bahamas Beach	FY94	307400	6
US	Bahamas Beach	FY95	376115	9
US	Hawaiian Club	FY93	479685	3
US	Hawaiian Club	FY94	519530	6
US	Hawaiian Club	FY95	480445	9

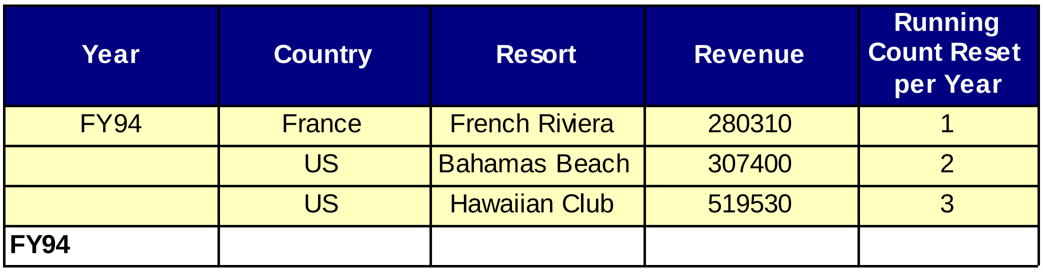
As with all Running Aggregates, you can calculate the cumulative decrease or increase of a numeric dimension or a measure per section or break.

The formula for this is :

=RunningCount(<Revenue>;<Year>)



Year	Country	Resort	Revenue	Running Count Reset per Year
FY93	France	French Riviera	295940	1
	US	Bahamas Beach	287929	2
	US	Hawaiian Club	479685	3
FY93				

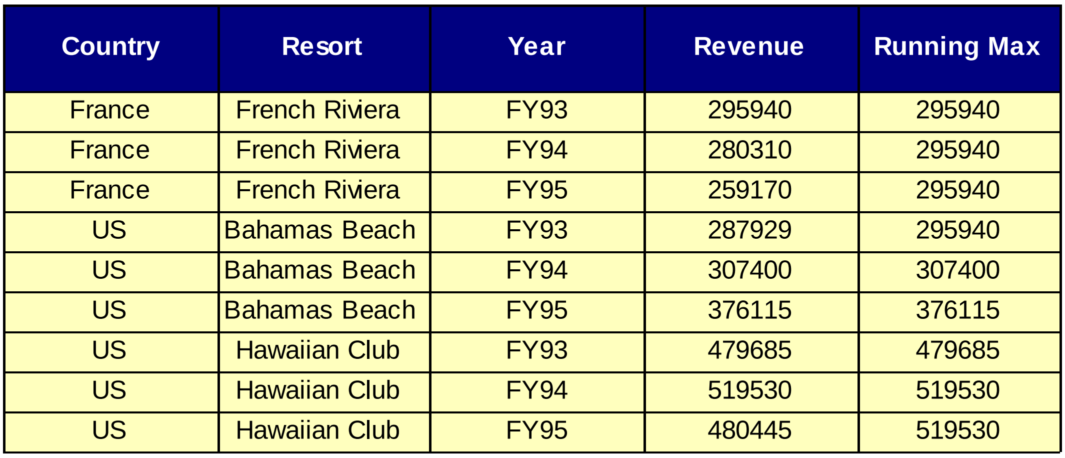


Year	Country	Resort	Revenue	Running Count Reset per Year
FY94	France	French Riviera	280310	1
	US	Bahamas Beach	307400	2
	US	Hawaiian Club	519530	3
FY94				

This applies to all Running Aggregates, when a decrease or increase is to be calculated per section or break.
When resetting the calculation per row or column in a crosstab the semi-colon should also be used. The dimension behind the semi-colon will be the object that specifies where the reset should take place. In the example shown on the previous page Year is the dimension for which the reset is taking place. The calculation will therefore restart from each first value per Year.

1.7 RunningMax

The RunningMax is an Aggregate Function that returns the running maximum of dimensional numeric values, or measures. For example: =RunningMax(<Revenue>) returns the running or cumulative maximum revenue.



Country	Resort	Year	Revenue	Running Max
France	French Riviera	FY93	295940	295940
France	French Riviera	FY94	280310	295940
France	French Riviera	FY95	259170	295940
US	Bahamas Beach	FY93	287929	295940
US	Bahamas Beach	FY94	307400	307400
US	Bahamas Beach	FY95	376115	376115
US	Hawaiian Club	FY93	479685	479685
US	Hawaiian Club	FY94	519530	519530
US	Hawaiian Club	FY95	480445	519530

As with all the Aggregates, the formula can be extended with the use of the Context Operators ForEach, ForAll, and In.
An example for this is:

=RunningMax(<Revenue>) In <Year>

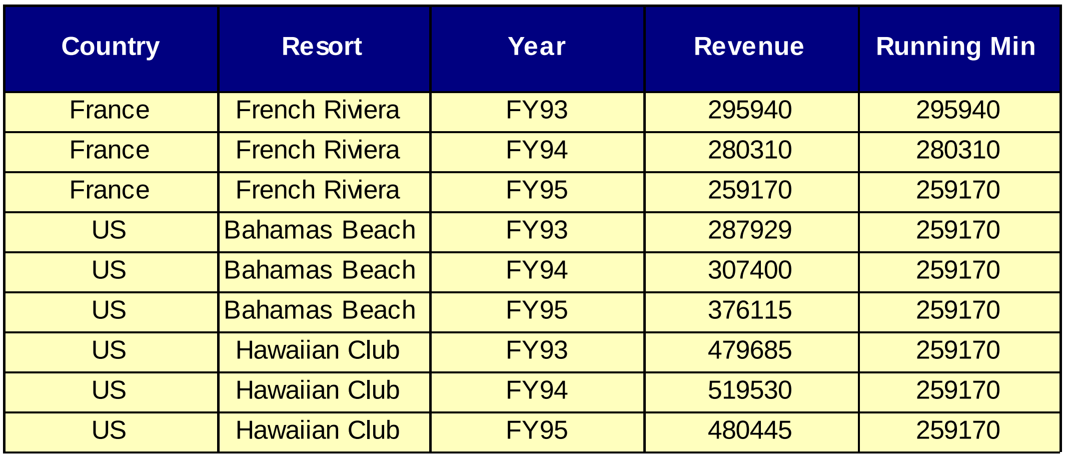
Also as with all Running Aggregates, you can calculate the cumulative decrease or increase of a numeric dimension or a measure per section or break.

The formula for this is:

=RunningMax(<Revenue>;<Year>)

1.8 RunningMin

The RunningMin is an Aggregate Function that returns the running minimum of dimensional numeric values, or measures. For example: =RunningMin(<Revenue>) returns the running or cumulative minimum revenue.



Country	Resort	Year	Revenue	Running Min
France	French Riviera	FY93	295940	295940
France	French Riviera	FY94	280310	280310
France	French Riviera	FY95	259170	259170
US	Bahamas Beach	FY93	287929	259170
US	Bahamas Beach	FY94	307400	259170
US	Bahamas Beach	FY95	376115	259170
US	Hawaiian Club	FY93	479685	259170
US	Hawaiian Club	FY94	519530	259170
US	Hawaiian Club	FY95	480445	259170

As with all the Aggregates, the formula can be extended with the use of the Context Operators ForEach, ForAll, and In.
An example for this is:

=RunningMin(<Revenue>) In <Year>

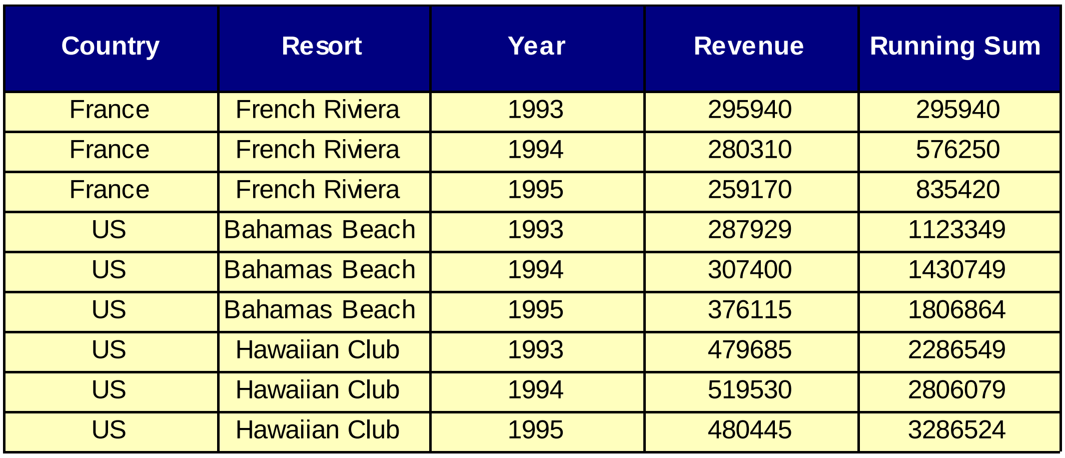
Also as with all Running Aggregates, you can calculate the cumulative decrease or increase of a numeric dimension or a measure per section or break.

The formula for this is:

=RunningMax(<Revenue>;<Year>)

1.9 RunningSum

The RunningSum will return a cumulative sum of the increase and decrease of a numeric dimension or a measure.
For example the formula: =RunningSum(<Revenue>) will return the cumulative sum of the revenue.



Country	Resort	Year	Revenue	Running Sum
France	French Riviera	1993	295940	295940
France	French Riviera	1994	280310	576250
France	French Riviera	1995	259170	835420
US	Bahamas Beach	1993	287929	1123349
US	Bahamas Beach	1994	307400	1430749
US	Bahamas Beach	1995	376115	1806864
US	Hawaiian Club	1993	479685	2286549
US	Hawaiian Club	1994	519530	2806079
US	Hawaiian Club	1995	480445	3286524

The first Revenue value will be repeated in the RunningSum column as it will be a calculation of: (295940+empty)=295940. The second value will be (295940+280310)=576250.The third value will be (295940+280310+259170)=835420.
(Note that the end figure of the RunningSum is the same figure as the Sum figure out the Insert Calculations menu. This is correct the total accumulative sum of the revenue would be the same as the total sum)

The extended syntax for this aggregate function allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

An Example for this is:

=RunningSum(<Revenue>) In <Country>

Country	Resort	Year	Revenue	RunningSum Per Country
France	French Riviera	1993	295940	835420
France	French Riviera	1994	280310	835420
France	French Riviera	1995	259170	835420
US	Bahamas Beach	1993	287929	3286524
US	Bahamas Beach	1994	307400	3286524
US	Bahamas Beach	1995	376115	3286524
US	Hawaiian Club	1993	479685	3286524
US	Hawaiian Club	1994	519530	3286524
US	Hawaiian Club	1995	480445	3286524

Also as with all Running Aggregates, you can calculate the cumulative decrease or increase of a numeric dimension or a measure per section or break.

For Example;

=RunningSum(<Revenue>;<Year>)

Year	Country	Resort	Revenue	RunningSum Reset per Year
1993	France	French Riviera	295940	295940
	US	Bahamas Beach	287929	583869
	US	Hawaiian Club	479685	1063554
1993				**1063554**

Year	Country	Resort	Revenue	RunningSum Reset per Year
1994	France	French Riviera	280310	280310
	US	Bahamas Beach	307400	587710
	US	Hawaiian Club	519530	1107240
1994				**1107240**

Another issue to look at with Running Aggregates, is when they are used in a crosstab.
In a crosstab, when a running aggregates is applied it will run from left to right (calculated per Row), and the last RunningSum value from the right-hand side of the column (835420), will be added to the first left-hand side value (287929) of the next column (making the first RunningSum value of the second row become 1123349), until the last value displayed in the crosstab. See report below.

		FY93	RunningSum	FY94	RunningSum	FY95	RunningSum
France	French Riviera	295940	295940	280310	576250	259170	835420
US	Bahamas Beach	287929	1123349	307400	1430749	376115	1806864
US	Hawaiian Club	479685	2286549	519530	2806079	480445	3286524

The actual calculation will be value for 93 first row + value for 94 first row + value for 95 first row + value for 93 second row + value for 94 second row, etc.
There are other possibilities to calculate the RunningSum in a crosstab, via the extended syntax Col, or Row. This will reset the value per row or per column.

For Example: =RunningSum(<Revenue>; Row<Resort>)

		FY93	RunningSum Reset per Column	FY94	RunningSum Reset per Column	FY95	RunningSum Reset per Column
France	French Riviera	295940	295940	280310	280310	259170	259170
US	Bahamas Beach	287929	583869	307400	587710	376115	635285
US	Hawaiian Club	479685	1063554	519530	1107240	480445	1115730

The first result would be (FY93)295940+empty =295940) , the second would be 295940+(FY94)280310=576250,the third would be 576250+(FY95)259170=835420.
Then it will reset to the next row i.e. it will start again from (FY93)287929+empty=287929,etc.

An example of the calculation per column is as follows:

=RunningSum(<Revenue>; Col <Year>)

		FY93	RunningSum Reset per Row	FY94	RunningSum Reset per Row	FY95	RunningSum Reset per Row
France	French Riviera	295940	295940	280310	280310	259170	259170
US	Bahamas Beach	287929	583869	307400	587710	376115	635285
US	Hawaiian Club	479685	1063554	519530	1107240	480445	1115730

The first result would be (FY93)295940+empty =295940) , the second would be 295940+(FY93)287929=583869,the third would be 583869+(FY93)479685=1063554.
Then it will reset to the next column i.e. it will start again from (FY94)280310+empty=280310,etc.

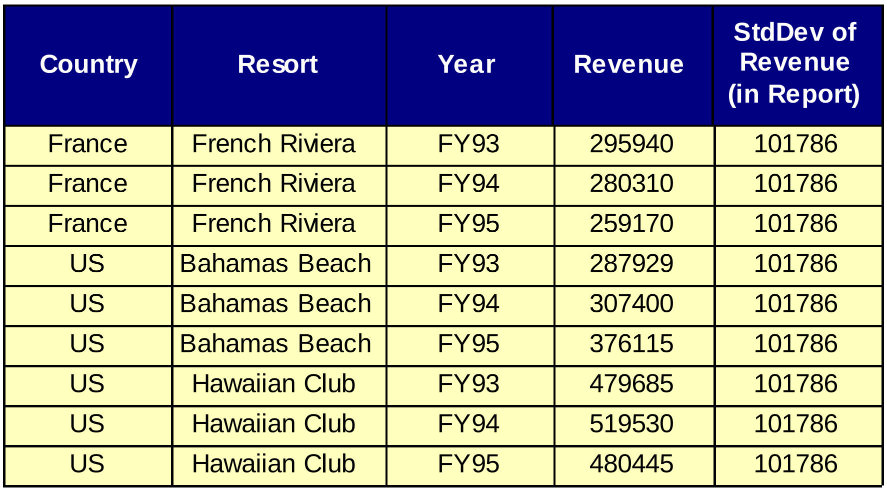
If the formula needs to be extended further, all that needs to be added is a comma, followed by a new dimensional object.
=RunningAggregate(<Measure>;<Dimension1>,<Dimension2>)

Note: the calculation per column or per row with the syntax Col, or Row can be used for all Running Aggregates, when using a crosstab, and is equivalent to:
=RunningAggregate(<Measure>;<Dimension>)

1.10 StdDev

The Function StdDev stands for Standard Deviation and calculates the standard deviation of a data sample.
Standard Deviation always concerns a range of numeric values, which can either be dimensional numeric values or measures.

For Example the Standard Deviation of the revenue:
=StdDev(<Revenue>) In Report
(NOTE: as this function is based on a range of data, when inserting this as a Column Value, you need to use In Report as an extended context to make the calculation based on the entire Revenue range in the table (report))



Country	Resort	Year	Revenue	StdDev of Revenue (in Report)
France	French Riviera	FY93	295940	101786
France	French Riviera	FY94	280310	101786
France	French Riviera	FY95	259170	101786
US	Bahamas Beach	FY93	287929	101786
US	Bahamas Beach	FY94	307400	101786
US	Bahamas Beach	FY95	376115	101786
US	Hawaiian Club	FY93	479685	101786
US	Hawaiian Club	FY94	519530	101786
US	Hawaiian Club	FY95	480445	101786

The Standard Deviation calculates the spread of the data range, by:

- Taking the average of the data range.
- Subtracting the average from each data sample (each revenue value minus the average of all the revenue values)
- Squaring all individual data samples

((value -average) *(value - average))

- Taking the sum from the squared data samples
- Dividing the sum of all squared data samples by the count of all data samples (N) minus 1.

Sum((value-average)*(value-average))/(N -1)

- Take the Square Root from the outcome of this.

Square Root((Sum((value-average)*(value-average))/(N -1))

- This will give the Standard Deviation value.

In the report the result will have been achieved as follows.

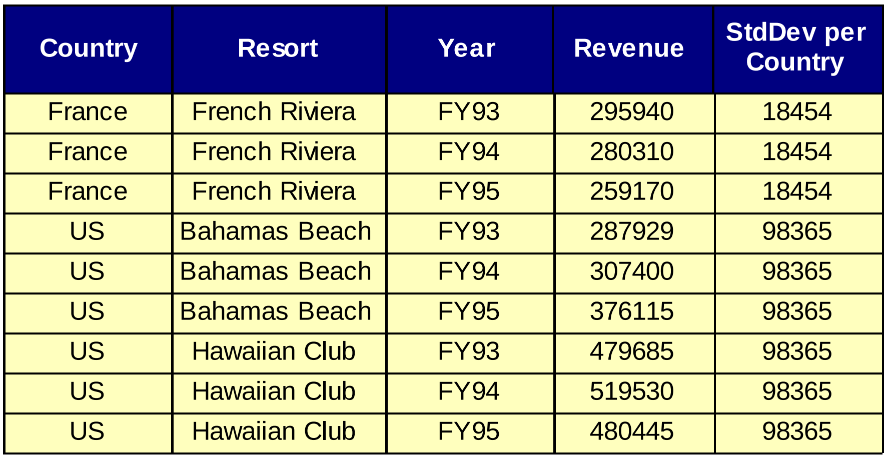
Revenue	Average of all Revenue Values	Rev - Ave	(Rev - Ave) * (Rev - Ave)
287929	365169	-77240	5966069093
295940	365169	-69229	4792700594
479685	365169	114516	13113837912
307400	365169	-57769	3337295874
280310	365169	-84859	7201106454
519530	365169	154361	23827215414
376115	365169	10946	119807619
259170	365169	-105999	11235858667
480445	365169	115276	13288479325
StdDev: 101786			**Sum: 82882370952**
Average: 365169			**Sum/(Count-1) : 10360296369**
Count: 9		**StdDev Result:**	**Sqrt(Sum/(Count-1)) : 101786**

Average of Revenue = 365169.
Calculate each Revenue Value - 365169.
Square each of these values: (Rev - Ave) * (Rev - Ave)
Sum this: 82882370952
Divide this sum by the (count of revenue - 1)
82882370952 / (9 -1) = 10360296369
Take the Square Root of this: Sqrt(10360296369) =101786, which is the Standard Deviation value.

Note: For StdDev you would use (N -1) as you deal with data samples. StdDevP would use (N) on its own as it deals with a calculation for the Population range.

The extended syntax for this Aggregate Function allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

For Example the Standard Deviation of the revenue per country:
StdDev(<Revenue>) In <Country>



Country	Resort	Year	Revenue	StdDev per Country
France	French Riviera	FY93	295940	18454
France	French Riviera	FY94	280310	18454
France	French Riviera	FY95	259170	18454
US	Bahamas Beach	FY93	287929	98365
US	Bahamas Beach	FY94	307400	98365
US	Bahamas Beach	FY95	376115	98365
US	Hawaiian Club	FY93	479685	98365
US	Hawaiian Club	FY94	519530	98365
US	Hawaiian Club	FY95	480445	98365

This will take into account the values per country as the value range, and you will have a StdDev value for each country.
Another Example: The calculation of the Standard Deviation of a data sample in country and year the formula would be as follows:
StdDev(<Revenue>) In (<Country>,<Year>)

NOTE: This function calculates the result set per row. This means that if for any reason a row gets taken from the table, e.g. when creating a summary table, the context used in the StdDev formula has to reflect the column(s) missing from the original table for which the calculation has been created, as otherwise the StdDev result will return empty cells.

For example:

You have created a report with Country, Year & Revenue. You have set Country as a break or section and have applied the StdDev on the Revenue (sum field) see report illustration below:

Country	Year	Revenue
France	FY93	295940
	FY94	280310
	FY95	259170
France	**StdDev:**	**18454**

Country	Year	Revenue
US	FY93	767614
	FY94	826930
	FY95	856560
US	**StdDev:**	**45291**

This shows the correct results as long as the Year column is present in the report. When this is removed the cells of the StdDev calculation will show up as blank cells:

Country	Revenue
France	835420
StdDev:	

Country	Revenue
US	2451104
StdDev:	

The reason is that the function StdDev calculates the results per row and it must keep the row reference it is to calculate within its context. To get the result displayed the context of the StdDev function has to become:
=StdDev(<Revenue> ForEach <Year>)
Which will bring back the correct result set:

Country	Revenue
France	835420
StdDev:	**18454**

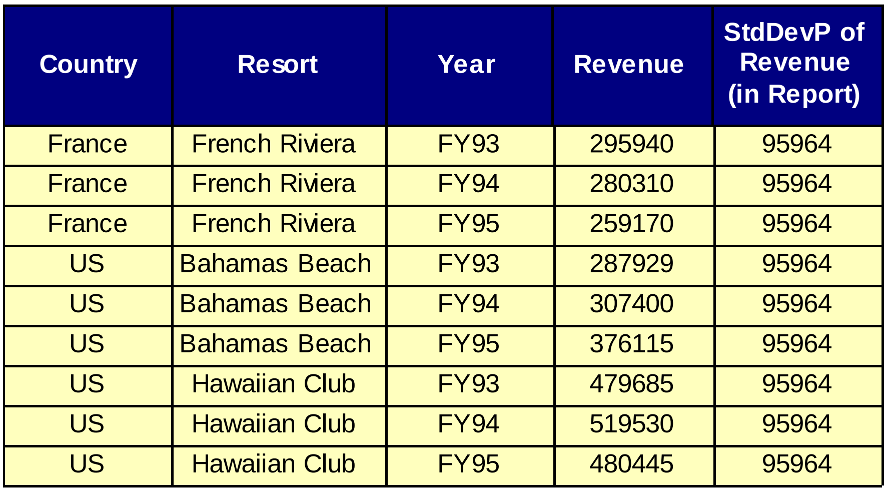
Country	Revenue
US	2451104
StdDev:	**45291**

Country	Revenue	StdDev ForEach Year
France	835420	18454
US	2451104	45291

1.11 StdDevP

The Function StdDevP stands for Standard Deviation Populous. It calculates the standard deviation of an entire data population. Standard Deviation always concerns a range of numeric values, which can either be dimensional numeric values or measures.

For Example the calculation of the entire data population of the Revenue:
StdDevP(<Revenue>) In Report
(NOTE: as this function is based on a range of data you need to use In Report as an extended context to make the calculation based on the entire Revenue range in the table (report))



Country	Resort	Year	Revenue	StdDevP of Revenue (in Report)
France	French Riviera	FY93	295940	95964
France	French Riviera	FY94	280310	95964
France	French Riviera	FY95	259170	95964
US	Bahamas Beach	FY93	287929	95964
US	Bahamas Beach	FY94	307400	95964
US	Bahamas Beach	FY95	376115	95964
US	Hawaiian Club	FY93	479685	95964
US	Hawaiian Club	FY94	519530	95964
US	Hawaiian Club	FY95	480445	95964

The Standard Deviation Populous calculates the spread of the data range of an entire population, by:

- Taking the average of the data range.
- Subtracting the average from each data sample (each revenue value minus the average of all the revenue values)
- Squaring all individual data samples

((value -average) *(value - average))

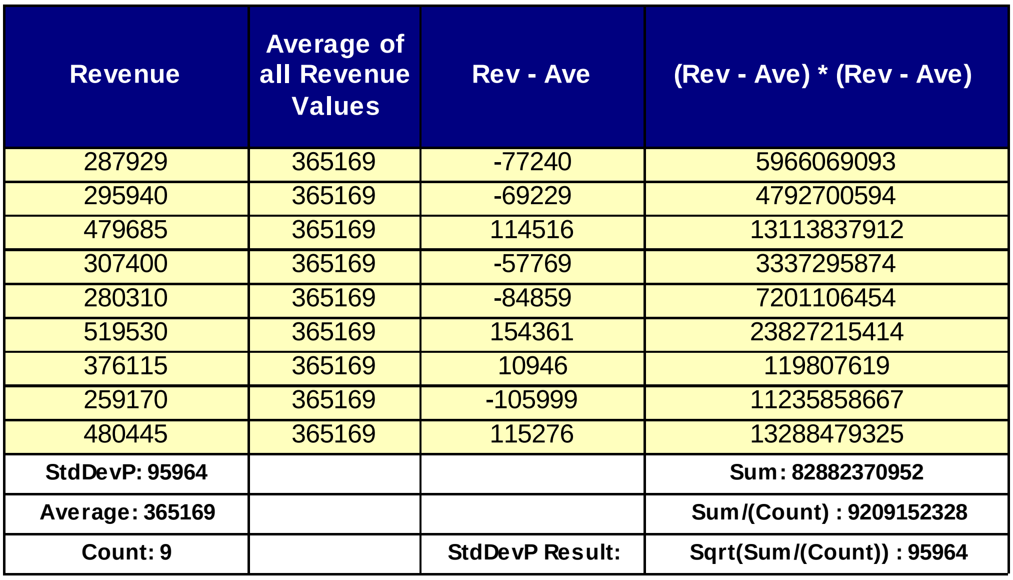
- Taking the sum from the squared data samples
- Dividing the sum of all squared data samples by the count of all data samples (N).

Sum((value-average)*(value-average))/(N)

- Take the Square Root from the outcome of this.

Square Root((Sum((value-average)*(value-average))/(N))
This will give the Standard Deviation Populous value.

In the report the result will have been achieved as follows.



Revenue	Average of all Revenue Values	Rev - Ave	(Rev - Ave) * (Rev - Ave)
287929	365169	-77240	5966069093
295940	365169	-69229	4792700594
479685	365169	114516	13113837912
307400	365169	-57769	3337295874
280310	365169	-84859	7201106454
519530	365169	154361	23827215414
376115	365169	10946	119807619
259170	365169	-105999	11235858667
480445	365169	115276	13288479325
StdDevP: 95964			**Sum: 82882370952**
Average: 365169			**Sum/(Count) : 9209152328**
Count: 9		**StdDevP Result:**	**Sqrt(Sum/(Count)) : 95964**

Average of Revenue = 365169.
Calculate each Revenue Value - 365169.
Square each of these values: (Rev - Ave) * (Rev - Ave)
Sum this: 82882370952
Divide this sum by the (count of revenue)
82882370952 / 9 = 9209152328
Take the Square Root of this: Sqrt(9209152328) =95964, which is the Standard Deviation value.

Note: For StdDev you would use (N -1) as you deal with data samples. StdDevP would use (N) on its own as it deals with a calculation for the Population range.

The extended syntax for this aggregate function allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

For Example the Standard Deviation of entire data population of the revenue per country:
StdDevP(<Revenue>) In <Country>

Country	Resort	Year	Revenue	StdDevP per Country
France	French Riviera	FY93	295940	15067
France	French Riviera	FY94	280310	15067
France	French Riviera	FY95	259170	15067
US	Bahamas Beach	FY93	287929	89795
US	Bahamas Beach	FY94	307400	89795
US	Bahamas Beach	FY95	376115	89795
US	Hawaiian Club	FY93	479685	89795
US	Hawaiian Club	FY94	519530	89795
US	Hawaiian Club	FY95	480445	89795

If extended further the calculation of the Standard Deviation for the entire data population in country and year the formula would be as follows:
=StdDevP(<Revenue>) In (<Country>,<Year>)

NOTE: This function calculates the result set per row. This means that if for any reason a row gets taken from the table, e.g. when creating a summary table, the context used in the StdDev formula has to reflect the column(s) missing from the original table for which the calculation has been created, as otherwise the StdDev result will return empty cells.

For example:

You have created a report with Country, Year & Revenue. You have set Country as a break or section and have applied the StdDevP on the Revenue (sum field) see report illustration below:

Country	Year	Revenue
France	FY93	295940
	FY94	280310
	FY95	259170
France	**StdDevP:**	**15067**

Country	Year	Revenue
US	FY93	767614
	FY94	826930
	FY95	856560
US	**StdDevP:**	**36980**

This shows the correct results as long as the Year column is present in the report. When this is removed the cells of the StdDevP calculation will show up as a result of 0:

Country	Revenue
France	835420
StdDevP:	**0**

Country	Revenue
US	2451104
StdDevP:	**0**

The reason is that the function StdDev calculates the results per row and it must keep the row reference it is to calculate within its context. To get the result displayed the context of the StdDev function has to become:
=StdDevP(<Revenue> ForEach <Year>)
Which will bring back the correct result set:

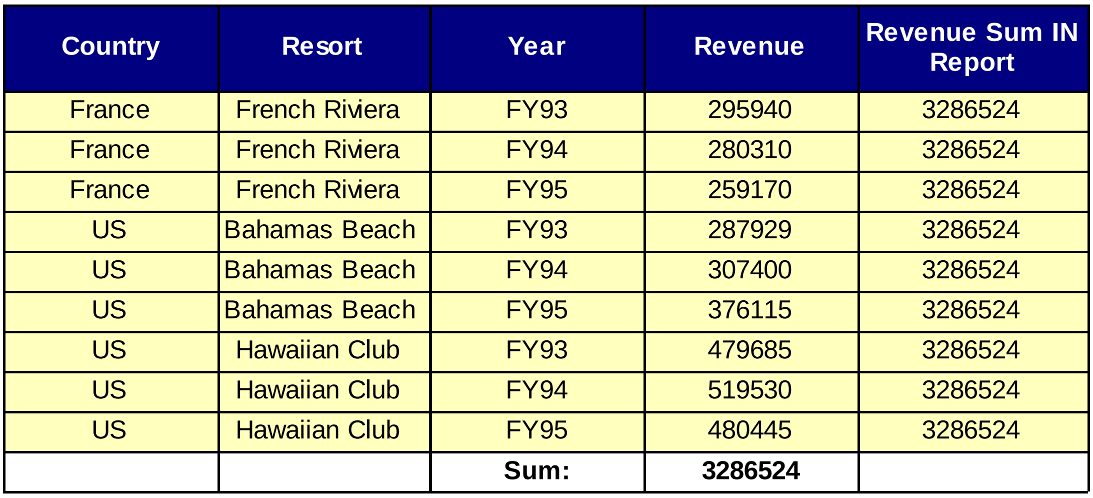
Country	Revenue
France	835420
StdDevP:	**15067**

Country	Revenue
US	2451104
StdDevP:	**36980**

Country	Revenue	StdDevP ForEach Year
France	835420	15067
US	2451104	36980

1.12 Sum

The Sum of an object can be obtained of any measure object or dimensional numeric value in the report. I have created an example here with the Sum of Revenue.
The Sum Function can be used in the same way as you would use the Sum Function in Insert Calculations. You can insert a column in the report and request the:
=Sum(<Revenue>) In Report.
In the report this will show the sum as 3286524.



Country	Resort	Year	Revenue	Revenue Sum IN Report
France	French Riviera	FY93	295940	3286524
France	French Riviera	FY94	280310	3286524
France	French Riviera	FY95	259170	3286524
US	Bahamas Beach	FY93	287929	3286524
US	Bahamas Beach	FY94	307400	3286524
US	Bahamas Beach	FY95	376115	3286524
US	Hawaiian Club	FY93	479685	3286524
US	Hawaiian Club	FY94	519530	3286524
US	Hawaiian Club	FY95	480445	3286524
		Sum:	**3286524**	

(Note: You would need the In Report as Business Objects calculates this per row. If it is excluded Business Objects will see all individual Revenue values as their own Sum.)

The extended syntax for Aggregate Functions allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.
An example for this with the Sum Function in this report will be to calculate the sum per dimension, for example Country.

This is also the way to deal with the total sum per Break or per Section.

The Formula used for this would be:
=Sum(<Revenue>) In <Country>

Country	Resort	Year	Revenue	Revenue Sum per Country
France	French Riviera	FY93	295940	835420
	French Riviera	FY94	280310	835420
	French Riviera	FY95	259170	835420
France				

Country	Resort	Year	Revenue	Revenue Sum per Country
US	Bahamas Beach	FY93	287929	2451104
	Bahamas Beach	FY94	307400	2451104
	Bahamas Beach	FY95	376115	2451104
	Hawaiian Club	FY93	479685	2451104
	Hawaiian Club	FY94	519530	2451104
	Hawaiian Club	FY95	480445	2451104
US				

Again this can be extended to the Sum per Country and Resort. The formula for this is:

=Sum(<Revenue>) In (<Country>,<Resort>)
Any further (dimensional) extensions can be added, separated by a comma.

Examples:

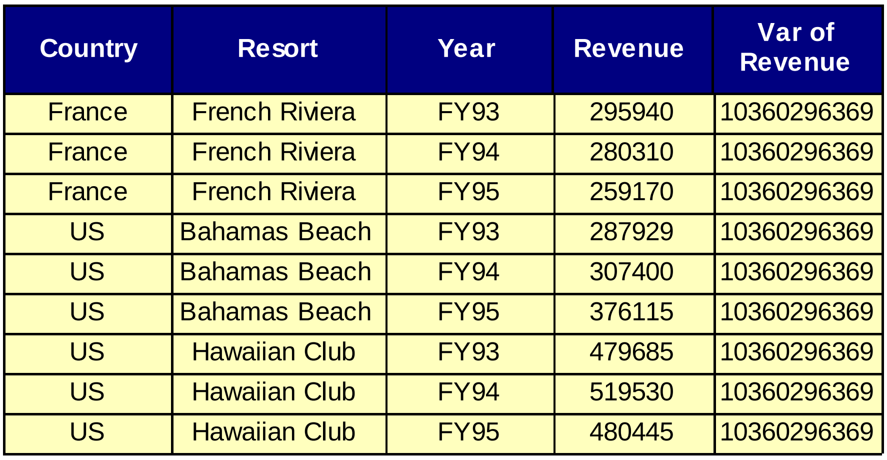
=Sum(<Revenue>) In (<Country>, <Resort>, <Year>)
=Sum(<Revenue> In (<Country>,<Resort>)) In (<Year>)

1.13 Var

The Function Var stands for Variance and calculates the *variance* (NOT to be confused with “variance” meaning “difference”) of a data range. It is in actual fact the Square of the Standard Deviation (StdDev * StdDev) of a data range.
The Variance Function always concerns a range of numeric values, which can either be dimensional numeric values or measures.

For Example the *variance* of the revenue
=Var(<Revenue>)In Report
(NOTE: as this function is based on a range of data you need to use In Report as an extended context to make the calculation based on the entire Revenue range in the table (report))

The result of the formula:
=(StdDev(<Revenue>) In Report)* (StdDev(<Revenue>) In Report)
would be the same as the result of the formula:
=Var(<Revenue>)In Report



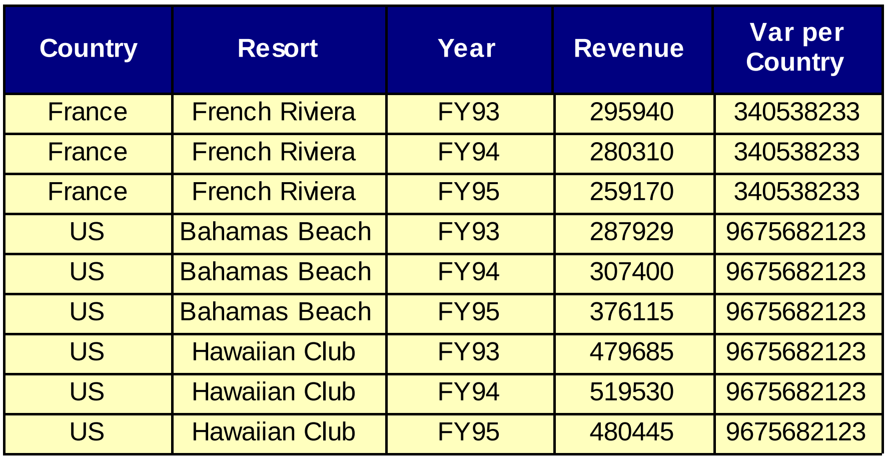
Country	Resort	Year	Revenue	Var of Revenue
France	French Riviera	FY93	295940	10360296369
France	French Riviera	FY94	280310	10360296369
France	French Riviera	FY95	259170	10360296369
US	Bahamas Beach	FY93	287929	10360296369
US	Bahamas Beach	FY94	307400	10360296369
US	Bahamas Beach	FY95	376115	10360296369
US	Hawaiian Club	FY93	479685	10360296369
US	Hawaiian Club	FY94	519530	10360296369
US	Hawaiian Club	FY95	480445	10360296369

Note: For Information on the calculation of the StdDev, check chapter 1.10 on StdDev)

The extended syntax for this aggregate function allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

For Example the *variance* of the revenue per country:

=Var(<Revenue>) In <Country>



Country	Resort	Year	Revenue	Var per Country
France	French Riviera	FY93	295940	340538233
France	French Riviera	FY94	280310	340538233
France	French Riviera	FY95	259170	340538233
US	Bahamas Beach	FY93	287929	9675682123
US	Bahamas Beach	FY94	307400	9675682123
US	Bahamas Beach	FY95	376115	9675682123
US	Hawaiian Club	FY93	479685	9675682123
US	Hawaiian Club	FY94	519530	9675682123
US	Hawaiian Club	FY95	480445	9675682123

If extended further the calculation of the *variance* of a data range in country and year the formula would be as follows:

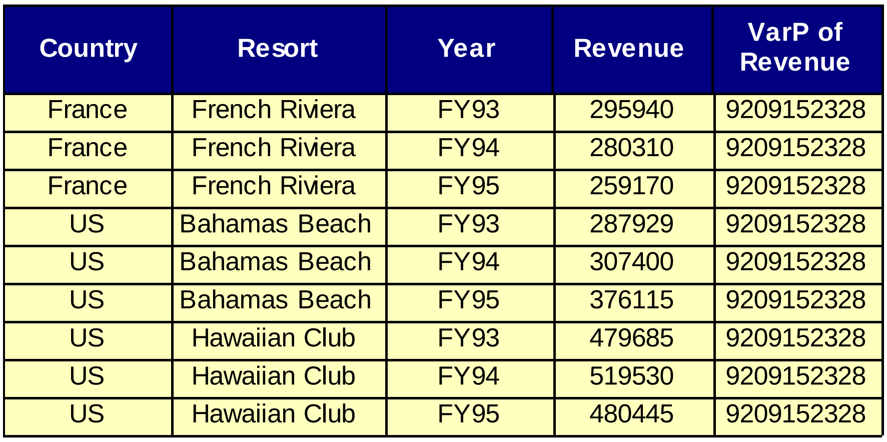
=Var(<Revenue>) In (<Country>,<Year>)

1.14 VarP

The Function VarP stands for Variance Populous and calculates the *variance* (NOT to be confused with “variance” meaning “difference”) of an entire data population. It is in actual fact the Square of the Standard Deviation Populous (StdDevP * StdDevP).

The Variance Populous Function always concerns a range of numeric values, which can either be dimensional numeric values or measures.
For Example the *variance* calculation of the entire data population of the Revenue:

=VarP(<Revenue>) In Report
(NOTE: as this function is based on a range of data you need to use In Report as an extended context to make the calculation based on the entire Revenue range in the table (report))



Country	Resort	Year	Revenue	VarP of Revenue
France	French Riviera	FY93	295940	9209152328
France	French Riviera	FY94	280310	9209152328
France	French Riviera	FY95	259170	9209152328
US	Bahamas Beach	FY93	287929	9209152328
US	Bahamas Beach	FY94	307400	9209152328
US	Bahamas Beach	FY95	376115	9209152328
US	Hawaiian Club	FY93	479685	9209152328
US	Hawaiian Club	FY94	519530	9209152328
US	Hawaiian Club	FY95	480445	9209152328

The result of the formula:
=(StdDevP(<Revenue>) In Report)* (StdDevP(<Revenue>) In Report) would give the same as the result of the formula:
=VarP(<Revenue>)In Report

Note: For Information on the calculation of the StdDevP, check chapter 1.11 on StdDevP)

The extended syntax for this Aggregate Function allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

For Example the *variance* of an entire data population of the revenue per country:

=VarP(<Revenue>) In <Country>

Country	Resort	Year	Revenue	VarP per Country
France	French Riviera	FY93	295940	227025489
France	French Riviera	FY94	280310	227025489
France	French Riviera	FY95	259170	227025489
US	Bahamas Beach	FY93	287929	8063068436
US	Bahamas Beach	FY94	307400	8063068436
US	Bahamas Beach	FY95	376115	8063068436
US	Hawaiian Club	FY93	479685	8063068436
US	Hawaiian Club	FY94	519530	8063068436
US	Hawaiian Club	FY95	480445	8063068436

If extended further the calculation of the *variance* of a an entire data population in country and year the formula would be as follows:

=VarP(<Revenue>) In (<Country>,<Year>)

2 Numeric functions

The Numeric Functions are divided into the following Functions:

Abs
Ceil
Cos
Exp
Fact
Floor
Ln
Log
Log10
Mod
Power
Rank
Round
ServerValue
Sign
Sin
Sqrt
Tan
ToNumber
Truncate

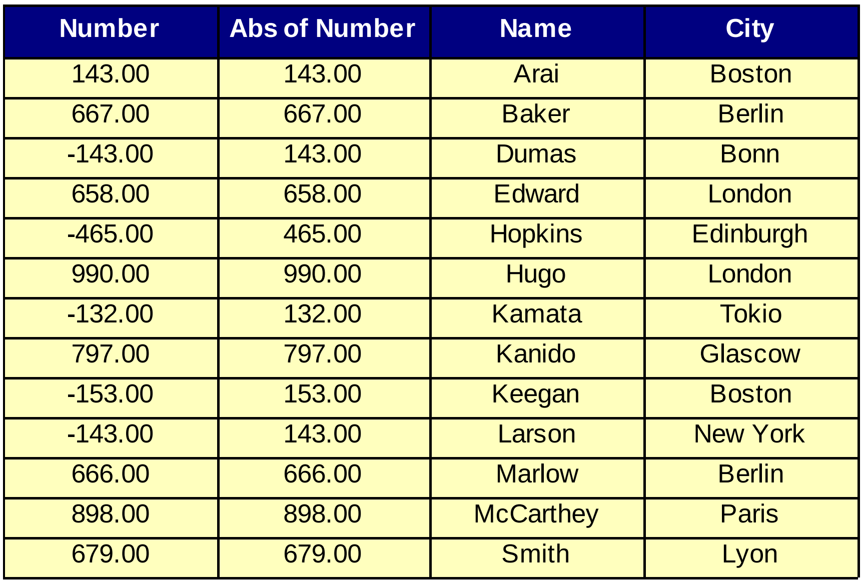
This Chapter will now have a close look at the individual Numeric Functions.

2.1 Abs

The Abs Function returns the absolute value of a number (a number’s *absolute* value is its number without its sign). This means that positive numbers may stay unchanged and negative numbers become positive numbers.

For Example:

=Abs(<number>)



Number	Abs of Number	Name	City
143.00	143.00	Arai	Boston
667.00	667.00	Baker	Berlin
-143.00	143.00	Dumas	Bonn
658.00	658.00	Edward	London
-465.00	465.00	Hopkins	Edinburgh
990.00	990.00	Hugo	London
-132.00	132.00	Kamata	Tokio
797.00	797.00	Kanido	Glascow
-153.00	153.00	Keegan	Boston
-143.00	143.00	Larson	New York
666.00	666.00	Marlow	Berlin
898.00	898.00	McCarthey	Paris
679.00	679.00	Smith	Lyon

Note that when the sum is taken of the absolute numbers this will take in account all the positive numbers that were negative before, and will therefore display a different total to the sum of the original number object.

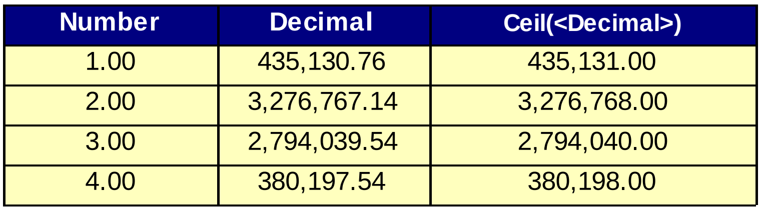
Also note that all Numeric Functions can be used in conjunction with other functions or objects.
For example: =Sum(Abs(<Number>)) In <City>
And all formulas can be set as a *Variable*

2.2 Ceil

The Ceil Function returns the value of a number rounded up to the nearest integer. This counts for all numerics (dimension or measure) were the number represents a value that is either positive, negative or null.

For example:

=Ceil(<decimal>)



Number	Decimal	Ceil(<Decimal>)
1.00	435,130.76	435,131.00
2.00	3,276,767.14	3,276,768.00
3.00	2,794,039.54	2,794,040.00
4.00	380,197.54	380,198.00

Note that all formula functions can be used in conjunction with other functions or objects.
For example: =Ceil(<decimal>) *<number>
And all formulas can be set as a *Variable*

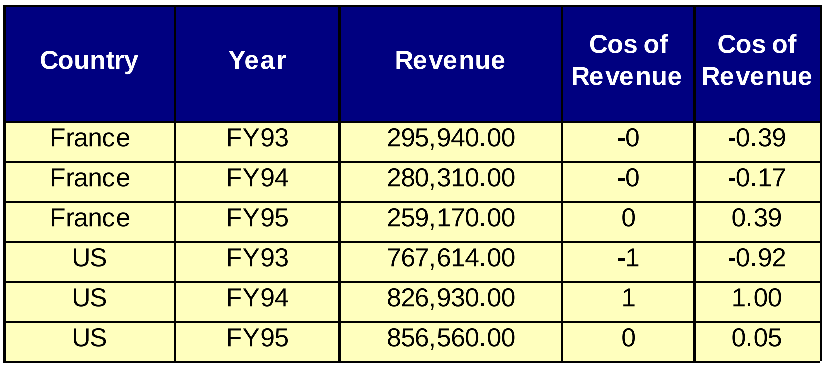
Also note that all Numeric Functions can be set based on an object or a single numeric value:
For example: Ceil (<decimal>), see report above
Or: Ceil(435,130.76) Which will only return 435,131.00

2.3 Cos

This Function stands for Cosine and calculates the cosine of the displayed number, returning the cosine of a number, where number is an angle in radians.

For Example:

=Cos(<Revenue>)



Country	Year	Revenue	Cos of Revenue	Cos of Revenue
France	FY93	295,940.00	-0	-0.39
France	FY94	280,310.00	-0	-0.17
France	FY95	259,170.00	0	0.39
US	FY93	767,614.00	-1	-0.92
US	FY94	826,930.00	1	1.00
US	FY95	856,560.00	0	0.05

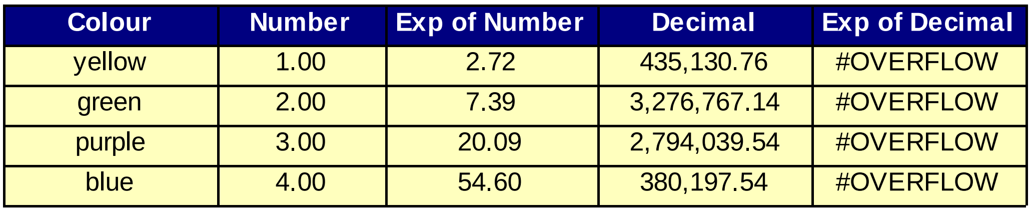
Be aware that the Cos Function will automatically round up to no decimal places if this format has been set in the report. In order to get the decimal places, the number format in Format Cell has to be set to two decimal places (0.00) (Note that decimal formatting also depends on the Regional Settings. Check if in the Regional Settings this has not been set to no decimal places.)

Also note that all Numeric Functions can be set based on an object or a single numeric value:
For example: =Cos (<Revenue>), see report above
Or: =Cos(213464) Which will only return 1

2.4 Exp

The Exp Function stands for a number exponent function and allows entry of scientific-notation numbers, returning ***e*** (equivalent to: 2,7182821828245) to the power of a number. It can be used on dimensional numeric values or measures.

For Example: Exp(3) = 20.09
Which is based on:
=2.718282182845*2.718282182845*2.718282182845



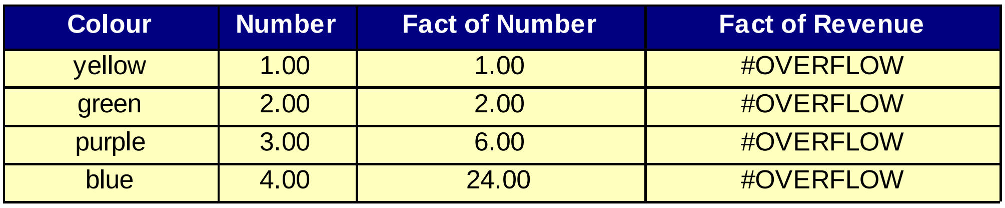
Colour	Number	Exp of Number	Decimal	Exp of Decimal
yellow	1.00	2.72	435,130.76	#OVERFLOW
green	2.00	7.39	3,276,767.14	#OVERFLOW
purple	3.00	20.09	2,794,039.54	#OVERFLOW
blue	4.00	54.60	380,197.54	#OVERFLOW

Be aware that the Exp function has a limit of 709.
If this is exceeded Business Objects will give an error of #Overflow (see column of Exp on Decimal) or crash with a General Page Fault (GPF).

Also note that all Numeric Functions can be set based on an object or a single numeric value.
For example: Exp (<Number>), see report above
Or: Exp(4) Which will only return 54.60

2.5 Fact

The Fact Function returns the factorial value of a dimensional numeric value or a measure.
For Example the when applying the Fact function on the <Number> column then the Fact(1) is equal to 1, the Fact (2) equal to 2, and the Fact(3) is equal to 6. The factorial value is the output value of the previous value times the new input value. With Fact(3) this is value 2 (previous output value) times 3 (new input value), resulting in the outcome of 6.



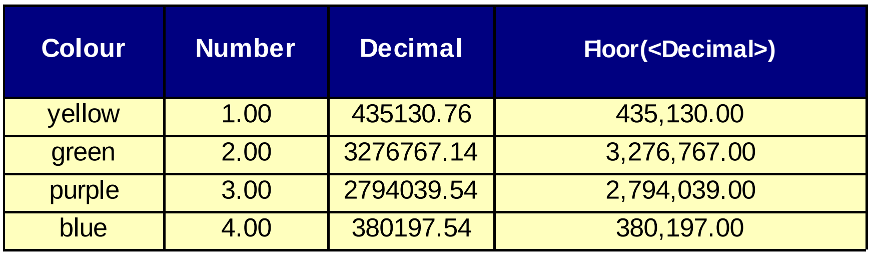
Colour	Number	Fact of Number	Fact of Revenue
yellow	1.00	1.00	#OVERFLOW
green	2.00	2.00	#OVERFLOW
purple	3.00	6.00	#OVERFLOW
blue	4.00	24.00	#OVERFLOW

Be aware that the Fact function has a limit of 170. If you exceed the Fact(170) this will return an error #OVERFLOW (see for example the Fact based on Decimal)

Also note that all Numeric Functions can be set based on an object or a single numeric value.
For example: Fact (<Number>), see report above
Or: Fact (4), which will only return 24.00

2.6 Floor

The Floor Function returns the value of a number rounded down to the nearest integer. This can be used on a dimensional numeric value or a measure.



Colour	Number	Decimal	Floor(<Decimal>)
yellow	1.00	435130.76	435,130.00
green	2.00	3276767.14	3,276,767.00
purple	3.00	2794039.54	2,794,039.00
blue	4.00	380197.54	380,197.00

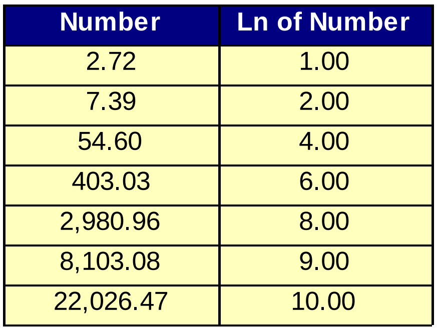
Be aware of formatting placed on the columns. The Decimal column has a format set as 0.00. The output of the Floor Function will automatically show the two decimal places as ".00" In order to remove this you need to alter the format in the Format Cell menu.
(Note that decimal formatting also depends on the Regional Settings. Check if in the Regional Settings this has not been set to no decimal places.)

As with all Numeric Functions the Floor function can be set based on an object or a single numeric value.
For example: Floor (<Number>), see report above
Or: Floor (2.5), which will only return 2.

2.7 Ln

The Ln Function returns the natural logarithm of the number. Natural logarithms are based on the constant **e** whose value is 2.71828182845904.

For Example the =Ln(7.39) returns 2.00 (based on 2.718281845904*2.718281845904=7.39)



Number	Ln of Number
2.72	1.00
7.39	2.00
54.60	4.00
403.03	6.00
2,980.96	8.00
8,103.08	9.00
22,026.47	10.00

As with all Numeric Functions the Ln Function can be set based on an object or a single numeric value:
For example: Ln (<Number>), see report above
Or: Ln (54.60), which will only return 4.00.

2.8 Log

The Log Function returns the logarithm of a real positive number to the specified base. =Log(number,base).

For Example the Log (8,2) returns 3
(as this is based on 2*2*2=8)

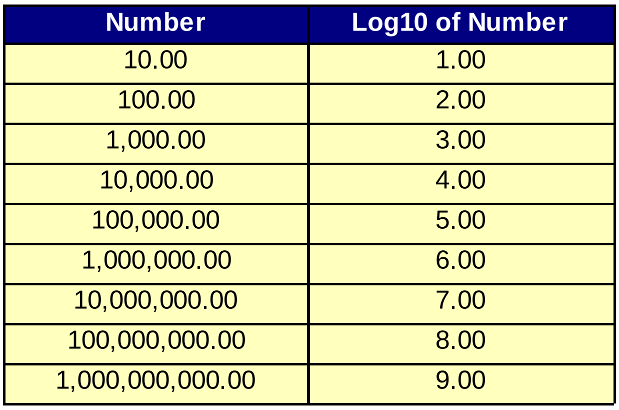
Number 1	Log of Number 1	Number 2	Log of Number 2
2.00	1.00	3.00	1.00
4.00	2.00	9.00	2.00
8.00	3.00	27.00	3.00
16.00	4.00	81.00	4.00
32.00	5.00	243.00	5.00
64.00	6.00	729.00	6.00
128.00	7.00	2,187.00	7.00
256.00	8.00	6,561.00	8.00
512.00	9.00	19,683.00	9.00

As with all Numeric Functions the Log Function can be set based on an object or a single numeric value.
For example: Log (<Number>, 2), see report above
Or: Log (16.00), which will only return 4.00 (based on 2*2*2*2)

2.9 Log10

The Log10 Function returns the logarithm of a real positive number to the base of 10.

For Example Log10(100) returns 2 as it is based on 10*10=100.



Number	Log10 of Number
10.00	1.00
100.00	2.00
1,000.00	3.00
10,000.00	4.00
100,000.00	5.00
1,000,000.00	6.00
10,000,000.00	7.00
100,000,000.00	8.00
1,000,000,000.00	9.00

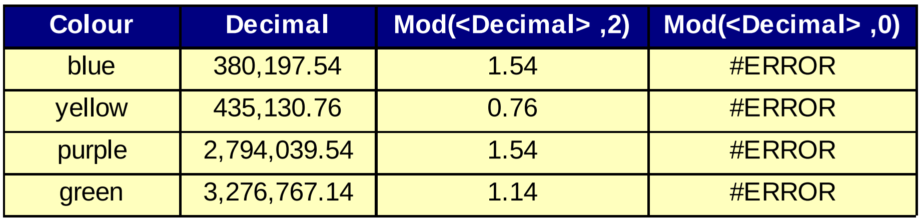
As with all Numeric Functions the Log10 Function can be set based on an object or a single numeric value.
For example: Log10 (<Number>), see report above
Or: =Log10 (1000.00), which will only return 3.00
(Based on 10*10*10=1000)

2.10 Mod

The Mod Function displays the modulus, or remainder of x/y. It returns the integer remainder of the division of the dividend number by the divisor number. Mod (dividend number, divisor number)

For Example: =Mod(435130.76,2) returns 0.76.

The result is found using the following formula:
=Mod(n, d) equals n-d* Floor(n/d)
In this case: 435130.76-2*(435130/2)=0.76



Colour	Decimal	Mod(<Decimal> ,2)	Mod(<Decimal> ,0)
blue	380,197.54	1.54	#ERROR
yellow	435,130.76	0.76	#ERROR
purple	2,794,039.54	1.54	#ERROR
green	3,276,767.14	1.14	#ERROR

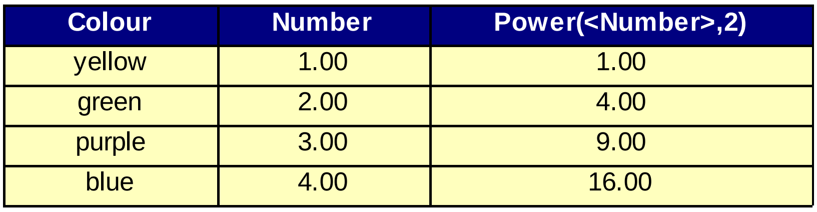
Note that the divisor number can not be set as zero, as Business Objects will return a #ERROR

Also note that as with all Numeric Functions the Mod Function can be set based on an object or a single numeric value.
For example: Mod(<Number>,2), see report above
Or: =Mod(380197.54 ,2) Which will only return 1.54

2.11 Power

The Power Function returns the value of a number to the specified power. =Power(number, power)

For Example: Power(3, 2) equals 9.

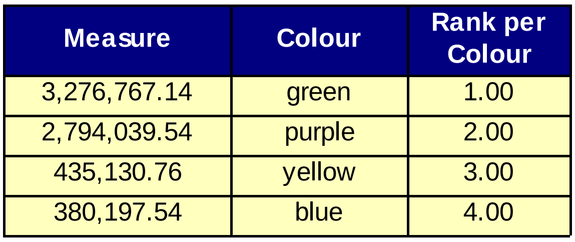


Colour	Number	Power(<Number>,2)
yellow	1.00	1.00
green	2.00	4.00
purple	3.00	9.00
blue	4.00	16.00

Note that as with all Numeric Functions the Power Function can be set based on an object or a single numeric value.
For example: =Power(<Number>,2), see report above
Or: =Power(4.00,2) Which will only return 16.00 (based on 4*4=16)

2.12 Rank

The Rank Function retrieves the rank number for each dimension value based on the measure.
=Rank(dimension, measure)



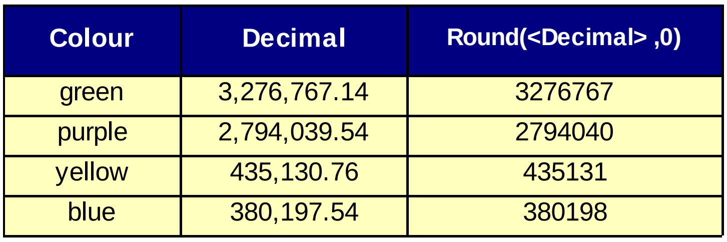
Measure	Colour	Rank per Colour
3,276,767.14	green	1.00
2,794,039.54	purple	2.00
435,130.76	yellow	3.00
380,197.54	blue	4.00

The Rank Function will set a number to a dimension according how it is ranked against the chosen measure.
In the example ‘Purple’ is set as number 1, as its measure value is the highest, and is therefore ranked as 1.

Note that the Rank function takes all structural filters into consideration, such as breaks or master/detail formatting.

2.13 Round

The Round Function will truncate a number to the specified number of decimal places, and rounds up the result to the nearest integer.
=Round(number, number of decimal places)



Colour	Decimal	Round(<Decimal> ,0)
green	3,276,767.14	3276767
purple	2,794,039.54	2794040
yellow	435,130.76	435131
blue	380,197.54	380198

Be aware of formatting placed on the columns. The Number column has a format set as 0.00. The output of the Round Function will automatically show the two decimal places as ".00" In order to remove this you need to alter the format in the Format Cell menu to the format of "0" for this column.
(Note that decimal formatting also depends on the Regional Settings. Check if in the Regional Settings this has not been set to no decimal places.)

Note that as with all Numeric Functions the Round Function can be set based on an object or a single numeric value.
For example: Round (<Decimal>), see report above
Or: =Round(156.87,0) Which will return 157.

2.14 ServerValue

The Server Value Function retrieves an aggregated value from an OLAP server.

Following is an illustration of server values.

	Q1	**Q2**	**Q3**	**Q4**
	Profit	**Profit**	**Profit**	**Profit**
East	41,855.00	83,725.00	124,408.00	70,596.00
West	37,715.00	81,006.00	116,808.00	64,493.00
Sum(<Profit>)	**79,570.00**	**164,731.00**	**241,216.00**	**135,089.00**
Sum(ServerValue<Profit>)	***121,398.00***	***254,972.00***	***206,362.00***	***927,375.00***

The values are retrieved for East and West, two of the four regions in the original data. These values (shown in Italics above) are defined as = ServerValue(<Profit>).

Two types of totals are shown:

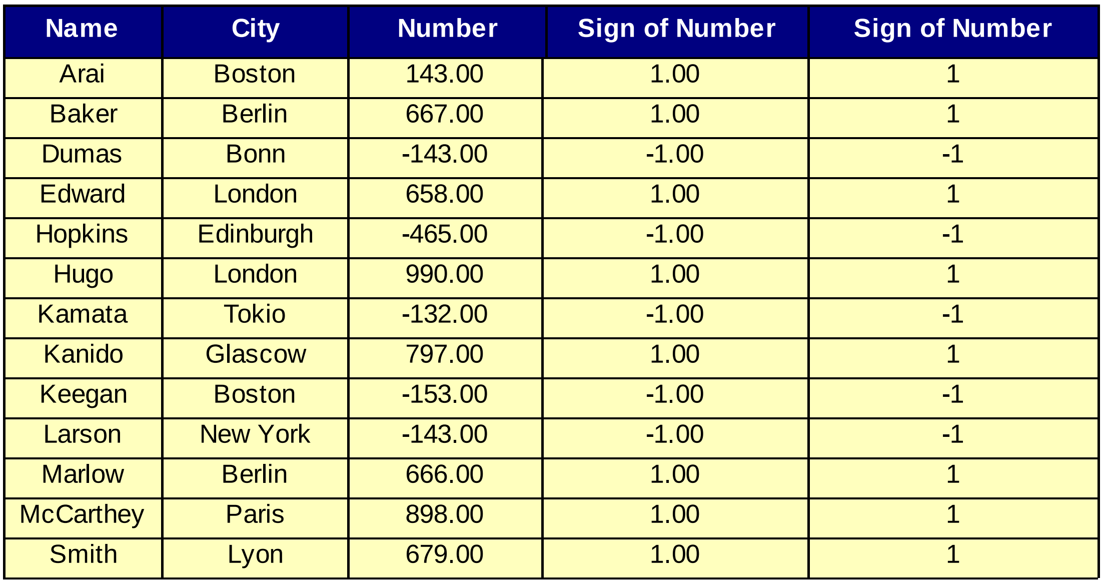
One: =Sum(<Profit>) is a local aggregate that sums the East/West values down columns and across rows (41,855 + 37,715 = 79,570, etc.)
Two: =Sum(ServerValue<Profit>), in contrast, is a ServerValue aggregate; this is the total of the four regions in the original OLAP server data, not just the two regions displayed in the report.

Note that Variable names for retrieved data are indicated with the additional notation <measure>(ServerValue), while local values are displayed as <measure>. If this function is used on a non-OLAP data provider (such as a universe-based data provider) then this function has no effect.

2.15 Sign

The Sign Function returns a number that represents the sign of the specified number (-1 a negative number, 0 for zero and 1 for a positive number.)

For Example: =Sign(-2543276) returns -1.



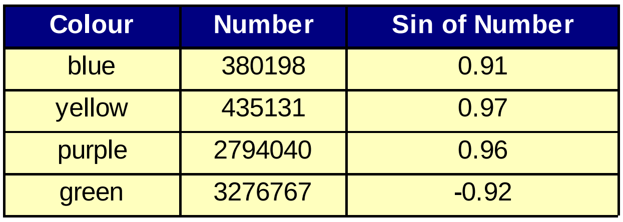
Name	City	Number	Sign of Number	Sign of Number
Arai	Boston	143.00	1.00	1
Baker	Berlin	667.00	1.00	1
Dumas	Bonn	-143.00	-1.00	-1
Edward	London	658.00	1.00	1
Hopkins	Edinburgh	-465.00	-1.00	-1
Hugo	London	990.00	1.00	1
Kamata	Tokio	-132.00	-1.00	-1
Kanido	Glascow	797.00	1.00	1
Keegan	Boston	-153.00	-1.00	-1
Larson	New York	-143.00	-1.00	-1
Marlow	Berlin	666.00	1.00	1
McCarthey	Paris	898.00	1.00	1
Smith	Lyon	679.00	1.00	1

Note that as with most Numeric Functions the Sign Function can be set based on an object or a single numeric value.
For example: =Sign(<Number>), see report above
Or: =Sign(990.00) Which will only return 1.

2.16 Sin

This Sin Function stands for Sine and calculates the sine of the displayed number, returning the sine of a number, where number is an angle in radians.

For Example: =Sin(234542) returns -0,116992.



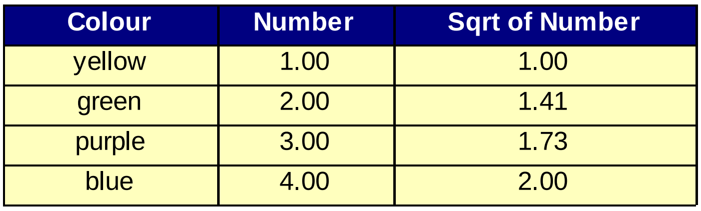
Colour	Number	Sin of Number
blue	380198	0.91
yellow	435131	0.97
purple	2794040	0.96
green	3276767	-0.92

Note that as with most Numeric Functions the Sin Function can be set based on an object or a single numeric value.
For example: =Sin(<Number>), see report above
Or: =Sin(380197.54) Which will only return 0.91.

2.19 Sqrt

The Function Sqrt stands for Square Root and will return the Square Root of a numeric value. The Square Root of a number is found by multiplying the number by itself

For Example: =Sqrt(16) returns 4 (based on 4*4=16)



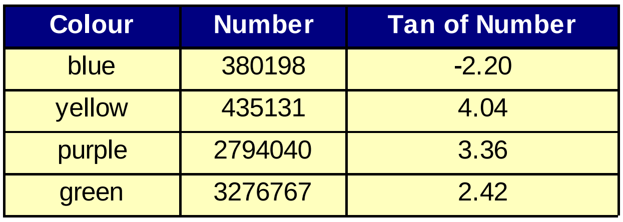
Colour	Number	Sqrt of Number
yellow	1.00	1.00
green	2.00	1.41
purple	3.00	1.73
blue	4.00	2.00

Note that as with most Numeric Functions the Sqrt Function can be set based on an object or a single numeric value.
For example: =Sqrt(<Number>), see report above
Or: =Sqrt(9.00), which will only return 3.00(based on 3*3=9).

2.20 Tan

This Tan function stands for Tangent and calculates the tangent of the displayed number, returning the tangent of a number, where number is an angle in radians.

For Example: Tan(380197.50) returns –2.46

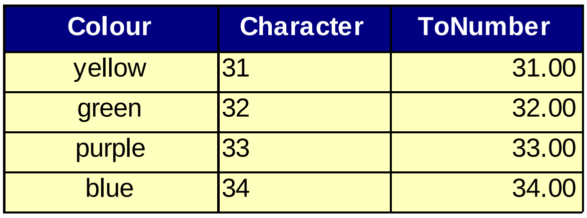


Colour	Number	Tan of Number
blue	380198	-2.20
yellow	435131	4.04
purple	2794040	3.36
green	3276767	2.42

Note that as with most Numeric Functions the Tan Function can be set based on an object or a single numeric value.
For example: =Tan(<Number>), see report above
Or: =Tan(435130.76), which will only return 4.04.

2.21 ToNumber

The Function ToNumber converts a character string to a number.



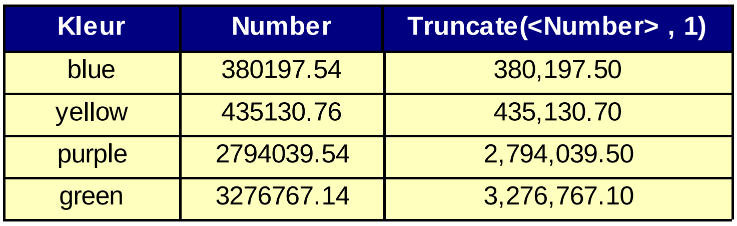
Colour	Character	ToNumber
yellow	31	31.00
green	32	32.00
purple	33	33.00
blue	34	34.00

Note that as with most Numeric Functions the ToNumber Function can be set based on an object or a single numeric value.
For example: =ToNumber(<Character>), see report above
Or: =ToNumber(“30”), which will return the numeric value 30.00.

2.22 Truncate

The Truncate Function truncates a number to the specified number of decimals. =Truncate(number, number of decimals)

For Example: =Truncate(128.35, 0), the result is 128.



Kleur	Number	Truncate(<Number> , 1)
blue	380197.54	380,197.50
yellow	435130.76	435,130.70
purple	2794039.54	2,794,039.50
green	3276767.14	3,276,767.10

Be aware of formatting placed on the columns. The Number column has a format set as 0.00. The output of the Truncate Function will automatically show the two decimal places as ".00",replacing the decimal it is truncating with a zero.
In order to make this show no zero you need to alter the format in the Format Cell menu to 0.0 for the truncation column.
(Note that decimal formatting also depends on the Regional Settings. Check if in the Regional Settings this has not been set to no decimal places.)

Note that as with most Numeric Functions the Truncate Function can be set based on an object or a single numeric value.
For example: =Truncate(<Number>,1), see report above
Or: =Truncate(435130.76, 1), which will only return 435130.70.

3 Character Functions

The Character Functions are divided into the following Functions:

Asc
Char
Concatenation
Fill
FormatDate
FormatNumber
InitCap
Left
LeftPad
LeftTrim
Length
Lower
Match
Pos
Replace
Right
RightPad
RightTrim
SubStr
Trim
Upper
WordCap

This Chapter will now have a close look at the individual Character Functions

3.1 Asc

The Asc Function returns an integer that corresponds to the ANSI code of the single character. =Asc("a single character")

For Example: =Asc("J") equals 74, as 74 is the ANSI code of the letter “J ” in upper case

Letter S	Asc of S	Letter s	Asc of s
S	83	s	115

The Asc Function is designed to return a code for each single character. When you insert the Asc(“France”) it will therefore only return the code equivalent to the first letter in the value. In this case the “F”, and will return the ANSI code 70 (which stands for the capital letter F)

Country	Asc of Country
Australia	65
France	70
US	85

Asc of U	Asc of F	Asc of A
85	70	65

Also note that letters in upper and lower case will return different ANSI codes, see report. The capital letter of S is equivalent to the ANSI code 83 and the lower case s is equivalent to ANSI code 115.

Note that as the function is asking for a character value, this needs to be accompanied with inverted commas.
The Formula is: =Asc(“F”)

3.2 Char

The Char Function returns a character, which is the number's ANSI value. =Char(number)

For Example: =Char(65) equals “A” (as 65 is the ASCII code for the letter “A”).

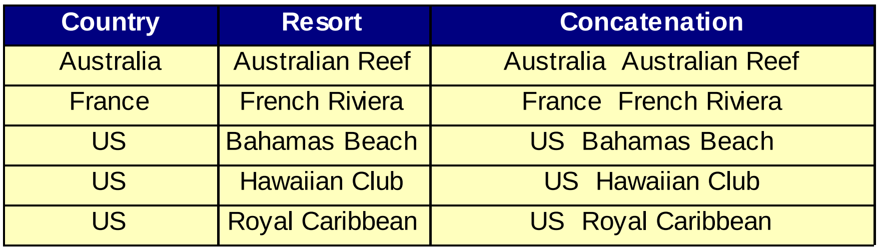
Number 70	Char of 70	Number 65	Char of 65
70	F	65	A

Note that the Char Function asks for a numeric value, and therefore does not need to have inverted commas.
The formula is: =Char(70)

3.3 Concatenation

The Concatenation Function concatenates two character strings. =Concatenation(character string, character string)

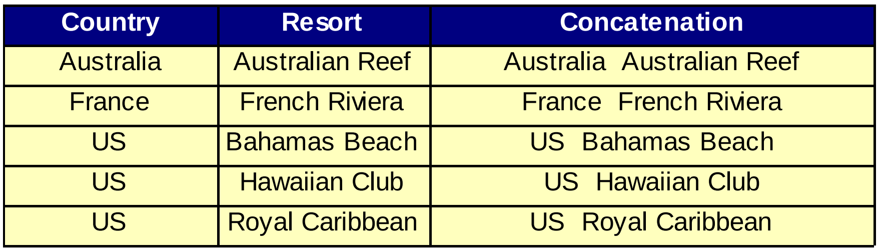
For Example: =Concatenation(<First Name>, <Last Name>) will return JohnSmith, where the value of First Name is John, and the value of Last Name, Smith.



Country	Resort	Concatenation
Australia	Australian Reef	Australia Australian Reef
France	French Riviera	France French Riviera
US	Bahamas Beach	US Bahamas Beach
US	Hawaiian Club	US Hawaiian Club
US	Royal Caribbean	US Royal Caribbean

The =Concatenation (<Country>,<Resort>) will return the result shown in the report above. The two values are set together without any spaces in between.
Please note that the Concatenation Function is equivalent to the operator "&", and will therefore enable you to concatenate two or more character strings, and to insert spaces between them.

For example: =Concatenation(<Country>&"   ",<Resort>)



Country	Resort	Concatenation
Australia	Australian Reef	Australia Australian Reef
France	French Riviera	France French Riviera
US	Bahamas Beach	US Bahamas Beach
US	Hawaiian Club	US Hawaiian Club
US	Royal Caribbean	US Royal Caribbean

This will provide the function with a space in between the two character strings. This can be taken further and more value strings can be added with further use of the &" "&
(Or: +" "+)operators.

For example: =Concatenation(<Country>&"   ",<Resort>&"   Text")
Which will return: France French Riviera Text

Please note that the Concatenation Function is designed only to concatenate character strings. If any non-character string is added to the function, Business Objects will display an error message ‘Incompatible data type’.

If however there is a need to concatenate with date and numeric objects, they need to be converted to a character string. This can be done via the Functions FormatDate and FormatNumber.

For example:

=Concatenation(<Country> , FormatDate(<Date>, "dd/mm/yyyy"))

Which will return: France12/2/1999

Or:

=Concatenation(<Country> , FormatNumber(<Number>, "0.00"))

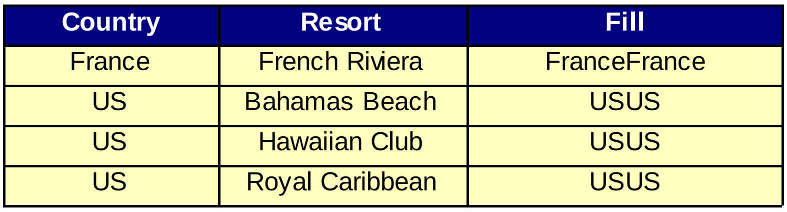
Which will return: France30.00

The Concatenation Function can be set with objects, for example =Concatenation(<Country>, <Resort>) , single values (=Concatenation(“France”,”US”)) or a combination of both.

3.4 Fill

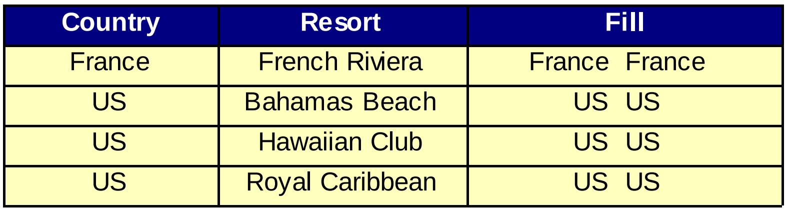
The Fill Function repeats a character string a certain number of times. =Fill(character string, number of times)

For example: =Fill(<Country>)", 2) returns :FranceFrance.



Country	Resort	Fill
France	French Riviera	FranceFrance
US	Bahamas Beach	USUS
US	Hawaiian Club	USUS
US	Royal Caribbean	USUS

Like with the Concatenation function the Fill function can be set with spaces via the &”   “& (or +”   “+) operators.



Country	Resort	Fill
France	French Riviera	France France
US	Bahamas Beach	US US
US	Hawaiian Club	US US
US	Royal Caribbean	US US

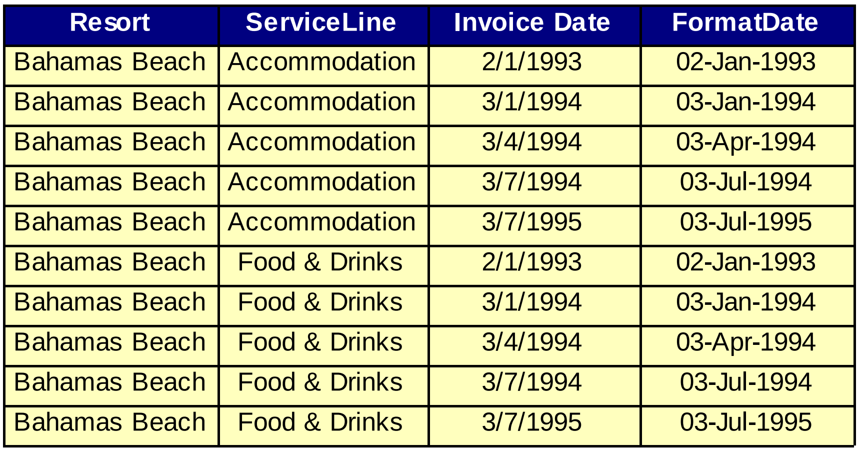
For example: =Fill(<Country>&"   " ,2)
This will return: France France

As with most Character Functions the Fill Function can be used with objects, Fill (<Country>, 2) see report, or with single character string values. For example: =Fill (“test”, 2) returning: “testtest”. Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.5 FormatDate

The FormatDate Function applies the string format to a date specified in standard dd/mm/yy format. It converts a date into the character string format required.
=FormatDate(date,"string format")

For example:
=FormatDate(<Date>, "mm/dd/yy") returns 07/29/96
=FormatDate('13/07/00', "m/d/yyyy") returns 7/13/2000



Resort	ServiceLine	Invoice Date	FormatDate
Bahamas Beach	Accommodation	2/1/1993	02-Jan-1993
Bahamas Beach	Accommodation	3/1/1994	03-Jan-1994
Bahamas Beach	Accommodation	3/4/1994	03-Apr-1994
Bahamas Beach	Accommodation	3/7/1994	03-Jul-1994
Bahamas Beach	Accommodation	3/7/1995	03-Jul-1995
Bahamas Beach	Food & Drinks	2/1/1993	02-Jan-1993
Bahamas Beach	Food & Drinks	3/1/1994	03-Jan-1994
Bahamas Beach	Food & Drinks	3/4/1994	03-Apr-1994
Bahamas Beach	Food & Drinks	3/7/1994	03-Jul-1994
Bahamas Beach	Food & Drinks	3/7/1995	03-Jul-1995

The FormatDate Function will allow you to change the date format that is set in the date object. If the date format in the date object is set to mm/yy, this can be changed with the FormatDate function to “mm/dd/yy”.
Do note that this will set the date to a character string.

As with most Character Functions the FormatDate Function can be used with objects,
=FormatDate(<Invoice Date>, “dd-mmm-yyyy”) see report , or with single date value taken from the date object (highlight the date object in the Edit Formula , and select ‘show list of values’)
.
For example: =FormatDate(“03/1/94”, “dd/mmm/yyyy”)
This will return the date value in the character string format that has been set in the formula: 03/Jan/1994

The FormatDate Function can also be used on other date functions available in the Edit Formula, such as CurrentDate /Document Date and LastExecutionDate. For Example: =FormatDate(CurrentDate(),”mm/dd/yy”)

Please note that the date format that is originally returned in the report is based on the Short Date Format in the Regional Settings.

FormatDate can also be handy when you have run a setup with more than one language, for example French and English. If you are swapping between languages, then this may cause an error with the dates. The reason being is that if you have run the report in English, you would have set the format to English settings =FormatDate(<date>,dd/mm/yyyy, and the French setting would need to be jj/mm/aaaa. When running this with BOLANGUAGE on EN then this will show the correct value, but when you change the BOLANGUAGE to FR this will give a #ERROR in the report. (Note that this BOLANGUAGE changes when you go to Tools / Options and change the Language you wish to run BO with from for example English to French – the BOLANGUAGE value will become FR.)

To solve this, you can make use of the FormatDate Function in conjunction with the Function: ApplicationValue.

The formula to use is:
=If ApplicationValue("BOLANGUAGE") ="FR" Then FormatDate(<date> , "jj/mm/aa") Else FormatDate(<date> ,"dd/mm/yy")

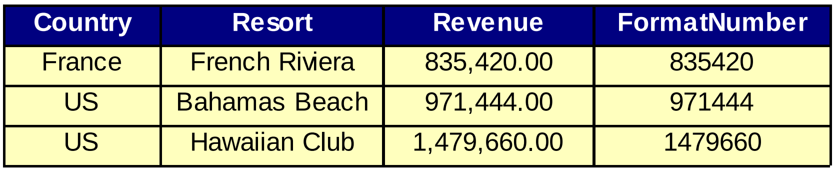
This will set the value correct, based on the language settings in the BOLANGUAGE file.

For more information on ApplicationValue see Misc Functions

3.6 FormatNumber

The FormatNumber Function applies a character string format to numbers. It allows you to convert a number to a character string.
=FormatNumber(number,"string format")

For example: =FormatNumber(12.5, "0.00") returns 12.50.



Country	Resort	Revenue	FormatNumber
France	French Riviera	835,420.00	835420
US	Bahamas Beach	971,444.00	971444
US	Hawaiian Club	1,479,660.00	1479660

The FormatNumber Function will allow you to change the number format that is set in the numeric object. If the number format in the numeric object is set to 12.00, this can be changed with the FormatNumber function to 12.
Do note that this will set the numeric value to a character string.

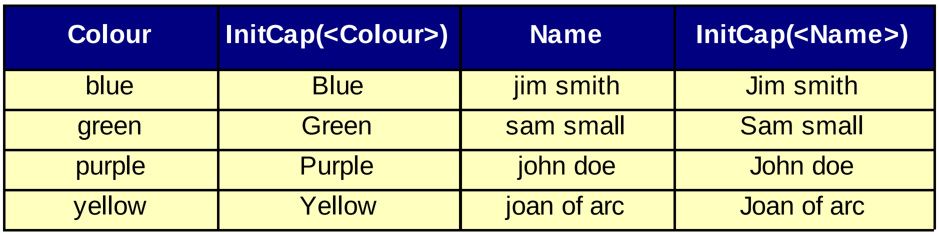
As with most Character Functions the FormatNumber Function can be used with objects, =FormatNumber(<Revenue>,“0”) see report , with a single number value taken from the numeric object (highlight the numeric object in the Edit Formula , and select ‘show list of values’) or a numeric value loose from any object.
For example: =FormatNumber(100.00,”0”) which will return the number without decimals and will return the value 100

Note that this function can also be used on other numeric functions available in the Edit Formula.
For example: =FormatNumber(Floor(6886133.95) ,"0")

3.7 InitCap

The Function InitCap will capitalise the first letter of a character string, while the remaining letters of stay in lower case. Also any words following will remain in lower case.
=InitCap(character string)

For example: =InitCap("john doe") returns John doe
(Note: for capitalising the first letter of all words, you need to use the Function WordCap)



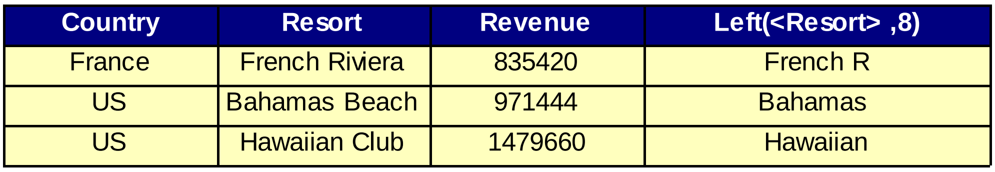
Colour	InitCap(<Colour>)	Name	InitCap(<Name>)
blue	Blue	jim smith	Jim smith
green	Green	sam small	Sam small
purple	Purple	john doe	John doe
yellow	Yellow	joan of arc	Joan of arc

As with most Character Functions the InitCap Function can be used with objects, =InitCap(<Names>) see report , or with single character string values. For example: =InitCap(“test”) returning: Test. Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.8 Left

The Left Function takes out a specified number of characters from the left of a character string. =Left(character string, number of characters)

For example: =Left("Date of the Order",4) returns Date.



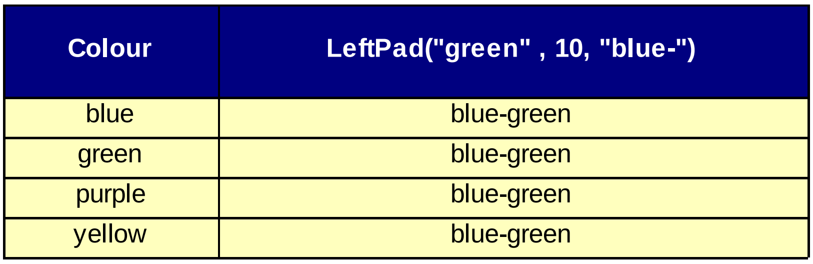
Country	Resort	Revenue	Left(<Resort> ,8)
France	French Riviera	835420	French R
US	Bahamas Beach	971444	Bahamas
US	Hawaiian Club	1479660	Hawaiian

As with most Character Functions the Left Function can be used with objects, =Left(<Resort>,8) see report , or with single character string values. For example: =Left(“today’s date”,5) returning: today. Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.9 LeftPad

The LeftPad function adds characters to the left of a character string. The total length of the string as is specified by number. =LeftPad(character string, number, character string)

For example: =LeftPad("port", 6, "re") returns report.



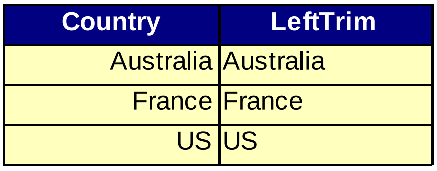
Colour	LeftPad("green" , 10, "blue-")
blue	blue-green
green	blue-green
purple	blue-green
yellow	blue-green

As with most Character Functions the Left Function can be used with objects, =LeftPad(<detail>,8,”text”) or with single character string values, as seen in the report ; =LeftPad("green" , 10 , "blue-") returning: blue-green.
Please note that when a single character string value is used, this always needs to be accompanied by inverted commas

3.10 LeftTrim

The LeftTrim Function deletes the blanks on the left of the character string (character string)

For example: =LeftTrim("   quarter") returns quarter.



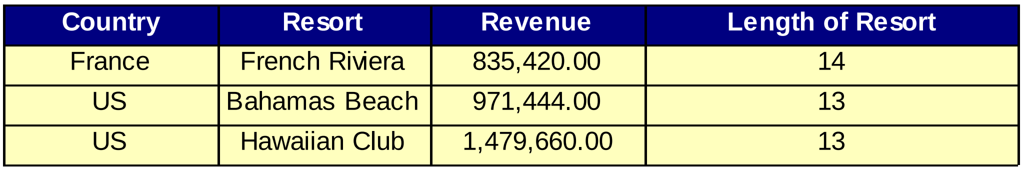
Country	LeftTrim
Australia	Australia
France	France
US	US

As with most Character Functions the Left Function can be used with objects, =LeftTrim(<country>) or with single character string values, as seen in the report ;=LeftTrim("   green") returning: green.
Please note that when a single character string value is used, this always needs to be accompanied by inverted commas

3.11 Length

The Length Function returns the number of characters in a character string. =Length (character string)

For example: =Length("SMITH") returns 5.



Country	Resort	Revenue	Length of Resort
France	French Riviera	835,420.00	14
US	Bahamas Beach	971,444.00	13
US	Hawaiian Club	1,479,660.00	13

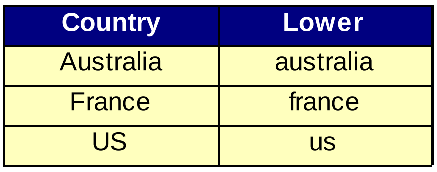
Please note that any spaces in the character string are counted within the Length function.

As with most Character Functions the Length Function can be used with objects, =Lenght(<Resort>),as seen in the report or with single character string values,=Lenght("France") returning:6.
Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.12 Lower

The Lower Function writes a character string in lower case.
=Lower(character string)

For example: =Lower("SMITH") returns smith.



Country	Lower
Australia	australia
France	france
US	us

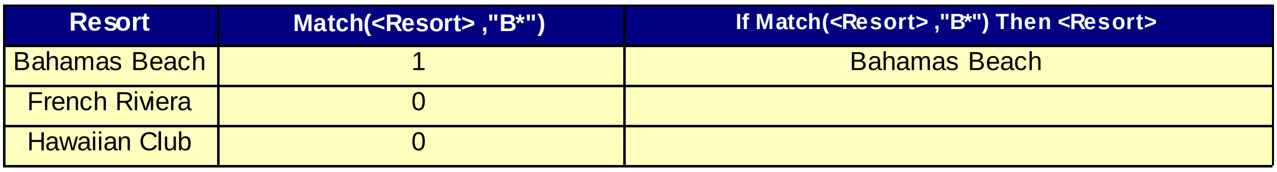
As with most Character Functions the Lower Function can be used with objects, =Lower(<Resort>),as seen in the report or with single character string values, =Lower("France") returning: france.
Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.13 Match

The Match Function compares two character strings;
=Match(character string, text pattern)

The Match Function is a logical function and will return 1 if the formula is true and 0 when false.
You use this function with an asterisk (*) and a question mark (?), which act as wildcard characters. The asterisk represents more than one character, while the question mark represents only one character.

For example: =Match(<Resort> ,"B*") will return a 1 for the resorts starting with a B and 0 for the others.
The Match function can be used with If Then Else.
For example: =If Match(<Resort>,"B*") Then "true" Else "false" returns true.



Resort	Match(<Resort> ,"B*")	If Match(<Resort> ,"B*") Then <Resort>
Bahamas Beach	1	Bahamas Beach
French Riviera	0	
Hawaiian Club	0	

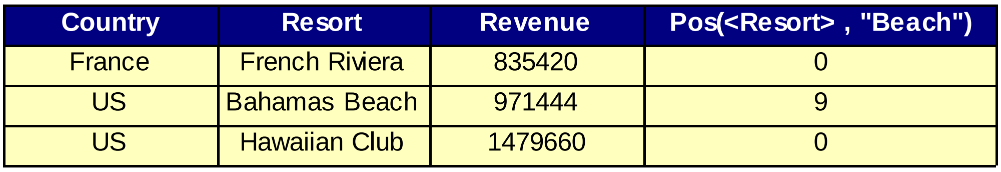
As with most Character Functions the Match Function can be used with objects, =Match(<Resort>,"B*"),as seen in the report or with single character string values, for example:
=If Match("SMITH", "S*") Then "true" Else "false" returns true.

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.14 Pos

The Pos Function returns the position of a character string within a longer character string.
=Pos(character string, text pattern)

For example: =Pos("Date of the Order","Date") returns 1, as Date is the first word in the character string.



Country	Resort	Revenue	Pos(<Resort> , "Beach")
France	French Riviera	835420	0
US	Bahamas Beach	971444	9
US	Hawaiian Club	1479660	0

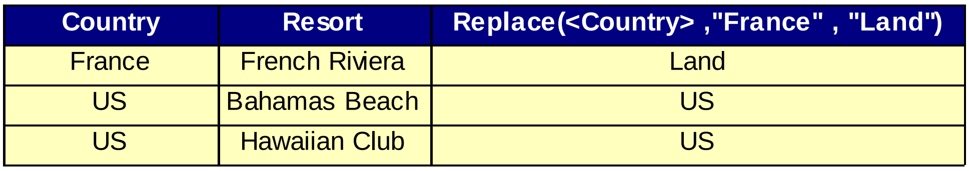
As with most Character Functions the Pos Function can be used with objects, =Pos(<Resort>,"Beach"),as seen in the report or with single character string values.

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.15 Replace

The Replace Function replaces part of a character string by another character string.
=Replace(character string, replace character string, new character string)

For example: =Replace("Paid Accommodation", "Paid", "Reserved") returns Reserved Accommodation instead of Paid Accommodation.



Country	Resort	Replace(<Country> ,"France" , "Land")
France	French Riviera	Land
US	Bahamas Beach	US
US	Hawaiian Club	US

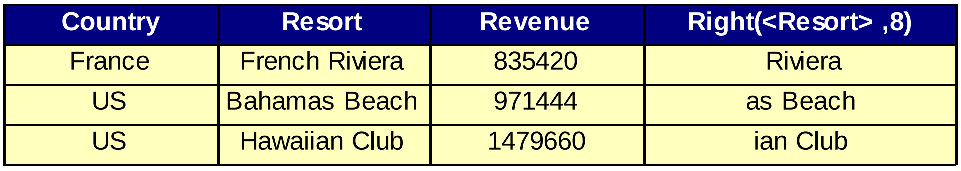
As with most Character Functions the Replace Function can be used with objects, =Replace(<Resort> ,<Resort> ,<Country>) as seen in the report or with single character string values.

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.16 Right

The Right Function takes out the specified number of characters from the character string.
=Right(character string, number of characters)

For example: =Right("Service Line", 5) returns Line.



Country	Resort	Revenue	Right(<Resort> ,8)
France	French Riviera	835420	Riviera
US	Bahamas Beach	971444	as Beach
US	Hawaiian Club	1479660	ian Club

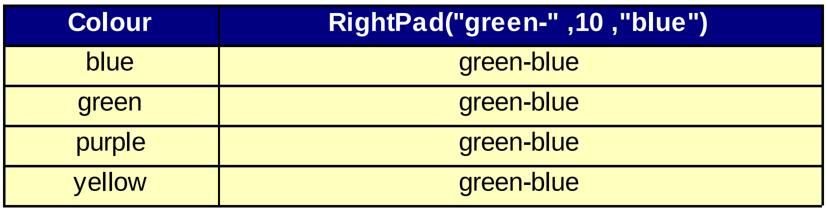
As with most Character Functions the Right Function can be used with objects, =Right(<Resort>,8) see report , or with single character string values. For example: =Right(“today’s date”,4) returning: date.

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.17 RightPad

The RightPad Function adds characters to the right of a character string. The total length of the string is specified by number.
=RightPad(character string, number, character string)

For example: =RightPad("Products", 15, " Loaned") returns Products Loaned.



Colour	RightPad("green-" ,10 ,"blue")
blue	green-blue
green	green-blue
purple	green-blue
yellow	green-blue

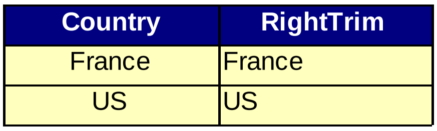
As with most Character Functions the RightPad Function can be used with objects, RightPad(<Colour>,8, “text”) or with single character string values, as seen in the report ;
=RightPad("green-" , 10 , "blue") returning: green-blue

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas

3.18 RightTrim

The RightTrim Function deletes the blanks on the right of a character string. =RightTrim(character string)

For example: =RightTrim("Smith   ") returns Smith.



Country	RightTrim
France	France
US	US

As with most Character Functions the RightTrim Function can be used with objects, =RightTrim(<country>) or with single character string values, as seen in the report ;=RightTrim(" green") returning: green.

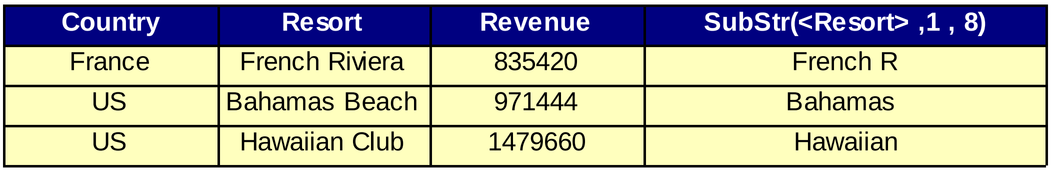
Please note that when a single character string value is used, this always needs to be accompanied by inverted commas

3.19 SubStr

The SubStr Function takes a sequence of characters from a character string.
=SubStr(character string, numberStart, numberLength)

For example: =SubStr("JAGUAR", 1, 3) is JAG.

The first number represents the position of the first character in the string that you want to take out. The second number represents the number of characters to extract, beginning with the first number. In the example, the first number is 1, which refers to the J in JAGUAR. The second number is 3, so the result is the JAG, the first three characters of JAGUAR.



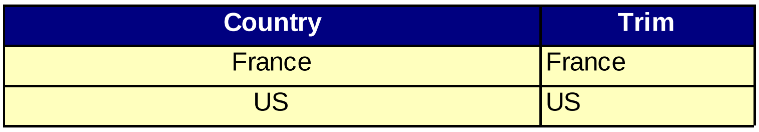
Country	Resort	Revenue	SubStr(<Resort> ,1 , 8)
France	French Riviera	835420	French R
US	Bahamas Beach	971444	Bahamas
US	Hawaiian Club	1479660	Hawaiian

As with most Character Functions the SubStr Function can be used with objects, =SubStr(<Resort>,1,6),as seen in the report or with single character string values.

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.20 Trim

The Trim Function deletes the blanks on either side of a character string. =Trim (character string)



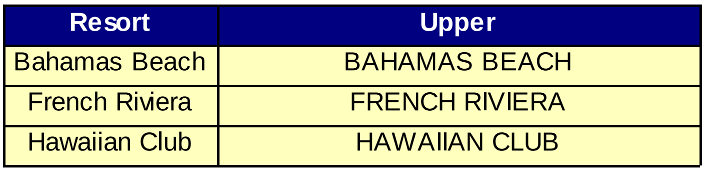
Country	Trim
France	France
US	US

For example: =Trim("   Great Britain   ") returns Great Britain. As with most Character Functions the Trim Function can be used with objects, or with single character string values for example Trim("   Country   ").

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.21 Upper

The Upper Function writes a character string in upper case. For example: =Upper("smith") returns SMITH. (character string)



Resort	Upper
Bahamas Beach	BAHAMAS BEACH
French Riviera	FRENCH RIVIERA
Hawaiian Club	HAWAIIAN CLUB

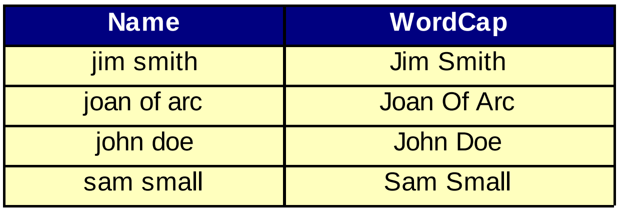
As with most Character Functions the Upper Function can be used with objects, =Upper(<Resort>),as seen in the report or with single character string values.

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

3.22 WordCap

The WordCap Function capitalises the first letter of each word in a character string. =WordCap(character string)

For example: =WordCap("smith and jones") returns Smith And Jones.



Name	WordCap
jim smith	Jim Smith
joan of arc	Joan Of Arc
john doe	John Doe
sam small	Sam Small

As with most Character Functions the WordCap Function can be used with objects, =WordCap(<Names>),as seen in the report or with single character string values.

Please note that when a single character string value is used, this always needs to be accompanied by inverted commas.

4 Date Functions

The Date Functions are divided into the following Functions:

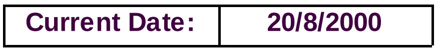
CurrentDate
CurrentTime
DayName
DayNumberofMonth
DayNumberOfWeek
DayNumberOfYear
DaysBetween
LastDayOfMonth
DayNumberOfWeek
Month
MonthNumberOfYear
MonthsBetween
Quarter
RelativeDate
ToDate
Week
Year

This Chapter will now continue to have a close look at each individual Date Function

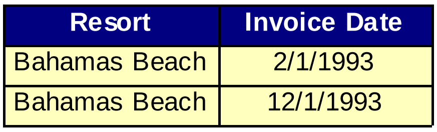
4.1 CurrentDate

The CurrentDate Function returns today's date

For example: =CurrentDate(), the result is today's date.



Current Date:	**20/8/2000**



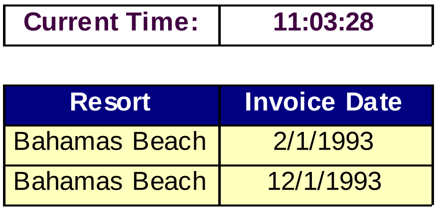
Resort	Invoice Date
Bahamas Beach	2/1/1993
Bahamas Beach	12/1/1993

Please note that the date format depends on the Regional Settings Short Date Format from the Regional Settings (in Settings/Control Panel/Regional Settings/Date Tab).
The format can also be set to a different format in the Number tab of the Cell Format dialog box.
Alternatively you can use the FormatDate Function to change the standard format to the required string format.

4.2 CurrentTime

The CurrentTime Function returns the current time

For example: =CurrentTime(), the result is the current time



Current Time:	**11:03:28**

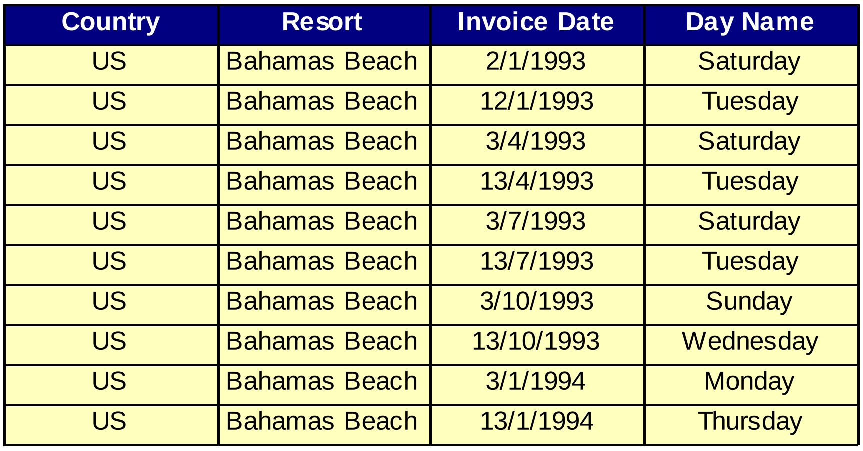
Resort	Invoice Date
Bahamas Beach	2/1/1993
Bahamas Beach	12/1/1993

The time format depends on the Regional Settings time style option (in Settings/Control Panel/Regional Settings/Time Tab), The format can also be set to a different format in the Number tab of the Cell Format dialog box.

4.3 DayName

The DayName Function returns the name of the day of the week in the date. =DayName(<Date>)

For example: =DayName(“01/14/1997”) returns Tuesday



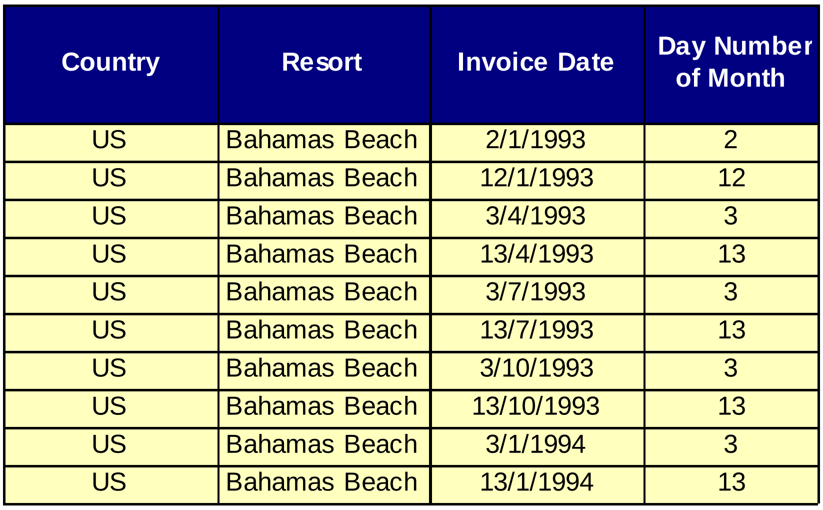
Country	Resort	Invoice Date	Day Name
US	Bahamas Beach	2/1/1993	Saturday
US	Bahamas Beach	12/1/1993	Tuesday
US	Bahamas Beach	3/4/1993	Saturday
US	Bahamas Beach	13/4/1993	Tuesday
US	Bahamas Beach	3/7/1993	Saturday
US	Bahamas Beach	13/7/1993	Tuesday
US	Bahamas Beach	3/10/1993	Sunday
US	Bahamas Beach	13/10/1993	Wednesday
US	Bahamas Beach	3/1/1994	Monday
US	Bahamas Beach	13/1/1994	Thursday

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on a single date value. (Note: This function corresponds with the language setting of your Regional Settings)

4.4 DayNumberofMonth

The DayNumberOfMonth Function returns the number of the day in the month. =DayNumberOfMonth(date)

For example: If the date-type variable Reservation Date returns 10/04/1995, DayNumberOfMonth(<Reservation Date>) returns 10.



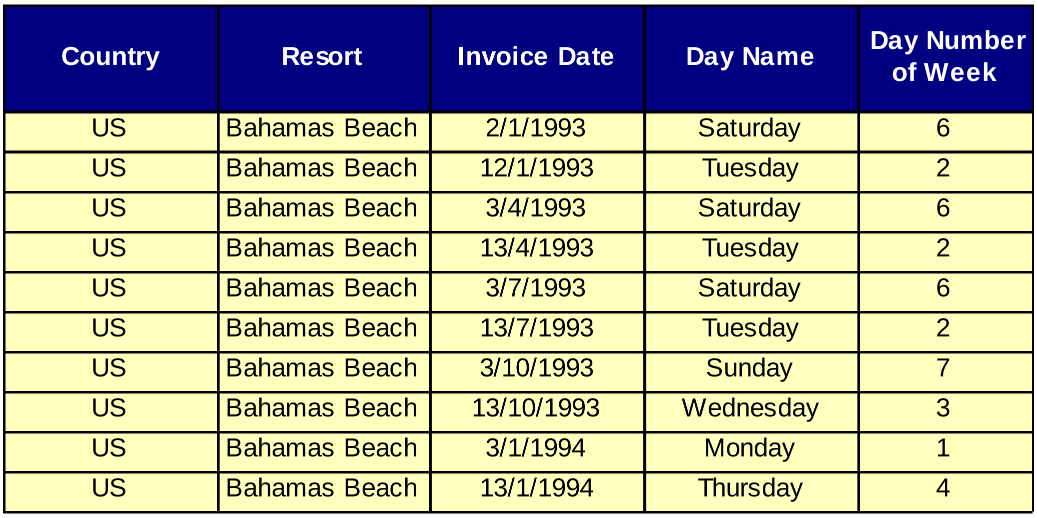
Country	Resort	Invoice Date	Day Number of Month
US	Bahamas Beach	2/1/1993	2
US	Bahamas Beach	12/1/1993	12
US	Bahamas Beach	3/4/1993	3
US	Bahamas Beach	13/4/1993	13
US	Bahamas Beach	3/7/1993	3
US	Bahamas Beach	13/7/1993	13
US	Bahamas Beach	3/10/1993	3
US	Bahamas Beach	13/10/1993	13
US	Bahamas Beach	3/1/1994	3
US	Bahamas Beach	13/1/1994	13

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on a single date value.

4.5 DayNumberOfWeek

The DayNumberOfWeek Function returns the number of the weekday in the date. =DayNumberOfWeek(date)

For example: If the date-type variable Reservation Date returns 10/04/1995, DayNumberOfWeek(<Reservation Date>) returns 1. (April 10 1995 was a Monday).



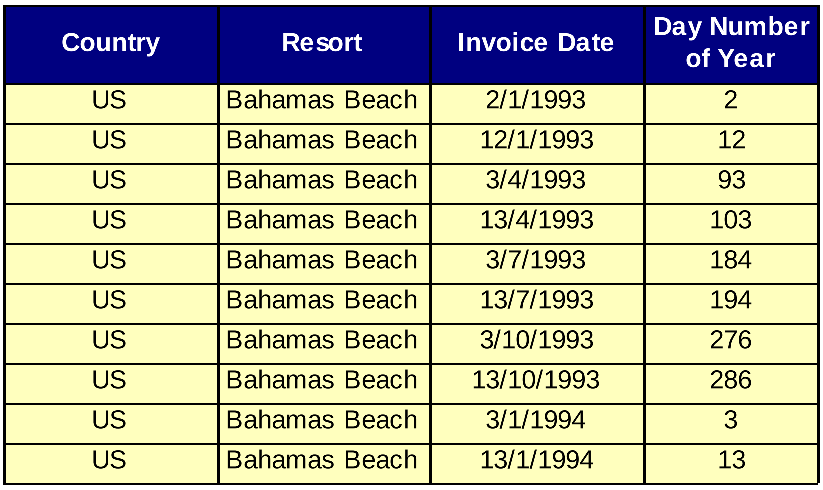
Country	Resort	Invoice Date	Day Name	Day Number of Week
US	Bahamas Beach	2/1/1993	Saturday	6
US	Bahamas Beach	12/1/1993	Tuesday	2
US	Bahamas Beach	3/4/1993	Saturday	6
US	Bahamas Beach	13/4/1993	Tuesday	2
US	Bahamas Beach	3/7/1993	Saturday	6
US	Bahamas Beach	13/7/1993	Tuesday	2
US	Bahamas Beach	3/10/1993	Sunday	7
US	Bahamas Beach	13/10/1993	Wednesday	3
US	Bahamas Beach	3/1/1994	Monday	1
US	Bahamas Beach	13/1/1994	Thursday	4

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on a single date value.

4.6 DayNumberOfYear

The DayNumberOfYear Function returns the number of the day in the year. =DayNumberOfYear(date)

For example: If the date-type variable Reservation Date returns 10/04/1995, DayNumberOfYear(<Reservation Date>) returns 100. (April 10 was the hundredth day of 1995.)



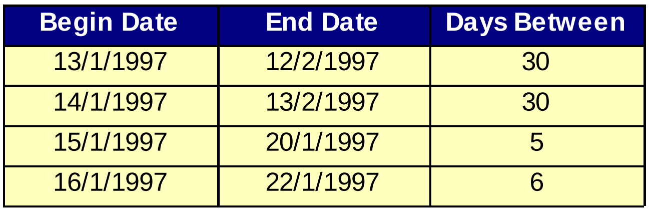
Country	Resort	Invoice Date	Day Number of Year
US	Bahamas Beach	2/1/1993	2
US	Bahamas Beach	12/1/1993	12
US	Bahamas Beach	3/4/1993	93
US	Bahamas Beach	13/4/1993	103
US	Bahamas Beach	3/7/1993	184
US	Bahamas Beach	13/7/1993	194
US	Bahamas Beach	3/10/1993	276
US	Bahamas Beach	13/10/1993	286
US	Bahamas Beach	3/1/1994	3
US	Bahamas Beach	13/1/1994	13

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on a single date value.

4.7 DaysBetween

The DaysBetween Function returns the number of days between two dates. =DaysBetween(date1, date2)

For example: =DaysBetween("10-04-94","12-04-94") returns 2.



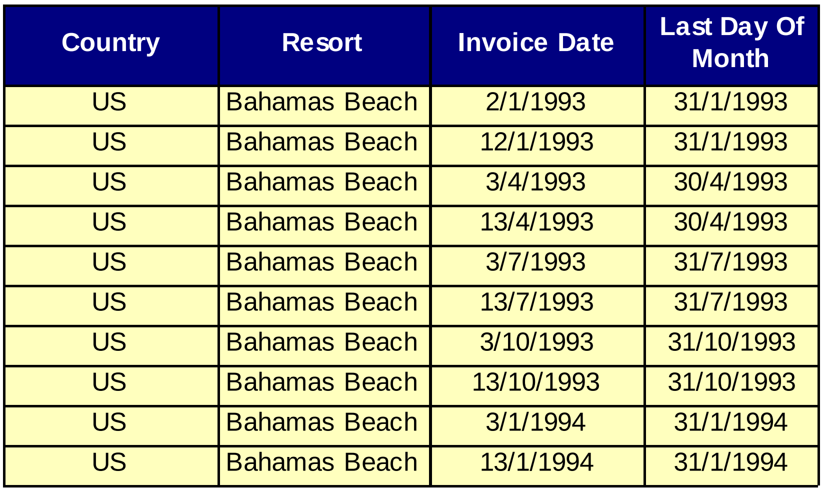
Begin Date	End Date	Days Between
13/1/1997	12/2/1997	30
14/1/1997	13/2/1997	30
15/1/1997	20/1/1997	5
16/1/1997	22/1/1997	6

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

4.8 LastDayOfMonth

The LastDayOfMonth Function returns the last day of the month in the date. =LastDayOfMonth(date)

For example: =LastDayOfMonth("08/05/1996") returns 31/05/1996.



Country	Resort	Invoice Date	Last Day Of Month
US	Bahamas Beach	2/1/1993	31/1/1993
US	Bahamas Beach	12/1/1993	31/1/1993
US	Bahamas Beach	3/4/1993	30/4/1993
US	Bahamas Beach	13/4/1993	30/4/1993
US	Bahamas Beach	3/7/1993	31/7/1993
US	Bahamas Beach	13/7/1993	31/7/1993
US	Bahamas Beach	3/10/1993	31/10/1993
US	Bahamas Beach	13/10/1993	31/10/1993
US	Bahamas Beach	3/1/1994	31/1/1994
US	Bahamas Beach	13/1/1994	31/1/1994

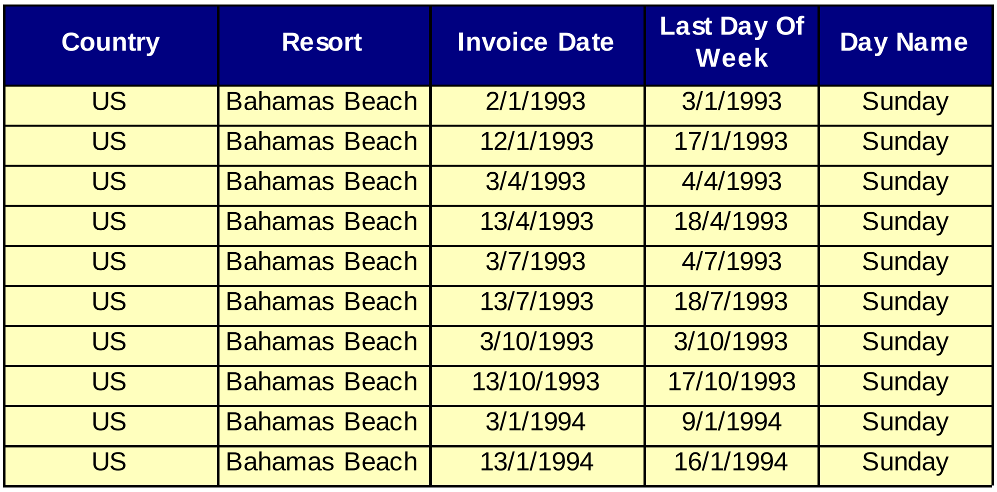
As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

Please note that the date format depends on the Regional Settings Short Date Format from the Regional Settings (in Settings/Control Panel/Regional Settings/Date Tab).
The format can also be set to a different format in the Number tab of the Cell Format dialog box.
Alternatively you can use the FormatDate Function to change the standard format to the required string format.

4.9 LastDayOfWeek

The LastDayOfWeek Function returns the date of the last day of the week in which the function's argument (date) occurs.
=LastDayOfWeek (date)

For example: =LastDayOfWeek("01/14/1997")returns:
01/19/1997



Country	Resort	Invoice Date	Last Day Of Week	Day Name
US	Bahamas Beach	2/1/1993	3/1/1993	Sunday
US	Bahamas Beach	12/1/1993	17/1/1993	Sunday
US	Bahamas Beach	3/4/1993	4/4/1993	Sunday
US	Bahamas Beach	13/4/1993	18/4/1993	Sunday
US	Bahamas Beach	3/7/1993	4/7/1993	Sunday
US	Bahamas Beach	13/7/1993	18/7/1993	Sunday
US	Bahamas Beach	3/10/1993	3/10/1993	Sunday
US	Bahamas Beach	13/10/1993	17/10/1993	Sunday
US	Bahamas Beach	3/1/1994	9/1/1994	Sunday
US	Bahamas Beach	13/1/1994	16/1/1994	Sunday

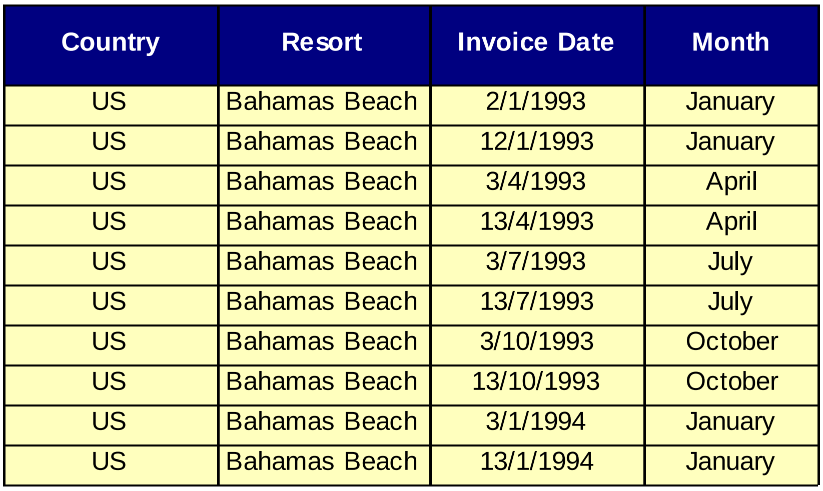
As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

Please note that the date format depends on the Regional Settings Short Date Format from the Regional Settings (in Settings/Control Panel/Regional Settings/Date Tab).
The format can also be set to a different format in the Number tab of the Cell Format dialog box.
Alternatively you can use the FormatDate Function to change the standard format to the required string format.

4.10 Month

The Month Function returns the name of the month in the date.
=Month(date)

For example: =Month("05/08/1995") returns: May



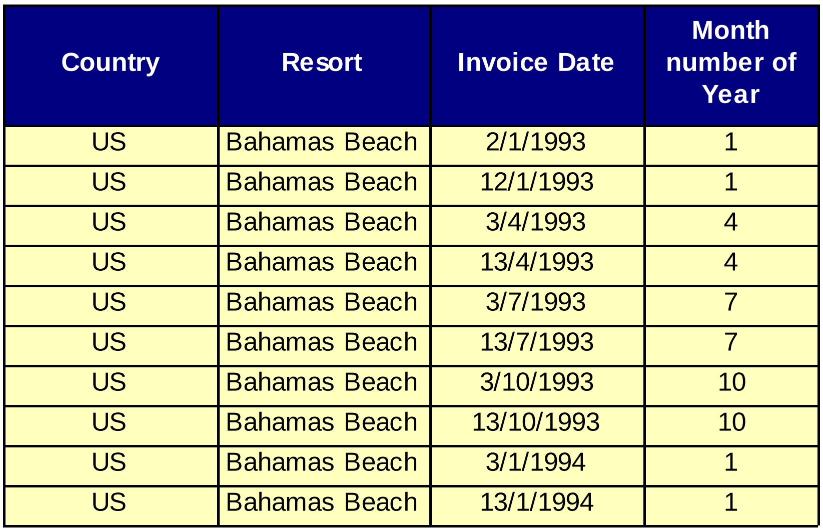
Country	Resort	Invoice Date	Month
US	Bahamas Beach	2/1/1993	January
US	Bahamas Beach	12/1/1993	January
US	Bahamas Beach	3/4/1993	April
US	Bahamas Beach	13/4/1993	April
US	Bahamas Beach	3/7/1993	July
US	Bahamas Beach	13/7/1993	July
US	Bahamas Beach	3/10/1993	October
US	Bahamas Beach	13/10/1993	October
US	Bahamas Beach	3/1/1994	January
US	Bahamas Beach	13/1/1994	January

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on a single date value. (Note: This function corresponds with the language setting of your Regional Settings)

4.11 MonthNumberOfYear

The MonthNumberOfYear Function returns the number of the month in the specified year. =MonthNumberOfYear (date)

For example: =Month("May 08 1996") returns 05 as May is the 5th month.



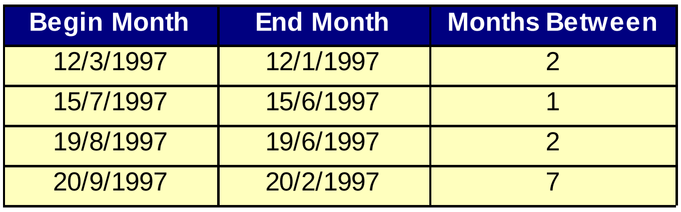
Country	Resort	Invoice Date	Month number of Year
US	Bahamas Beach	2/1/1993	1
US	Bahamas Beach	12/1/1993	1
US	Bahamas Beach	3/4/1993	4
US	Bahamas Beach	13/4/1993	4
US	Bahamas Beach	3/7/1993	7
US	Bahamas Beach	13/7/1993	7
US	Bahamas Beach	3/10/1993	10
US	Bahamas Beach	13/10/1993	10
US	Bahamas Beach	3/1/1994	1
US	Bahamas Beach	13/1/1994	1

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on a single date value.

4.12 MonthsBetween

The MonthsBetween Function returns the number of months between two dates. =MonthsBetween('date1', 'date2')

For example: =MonthsBetween('10-04-94', '10-05-94') returns 1



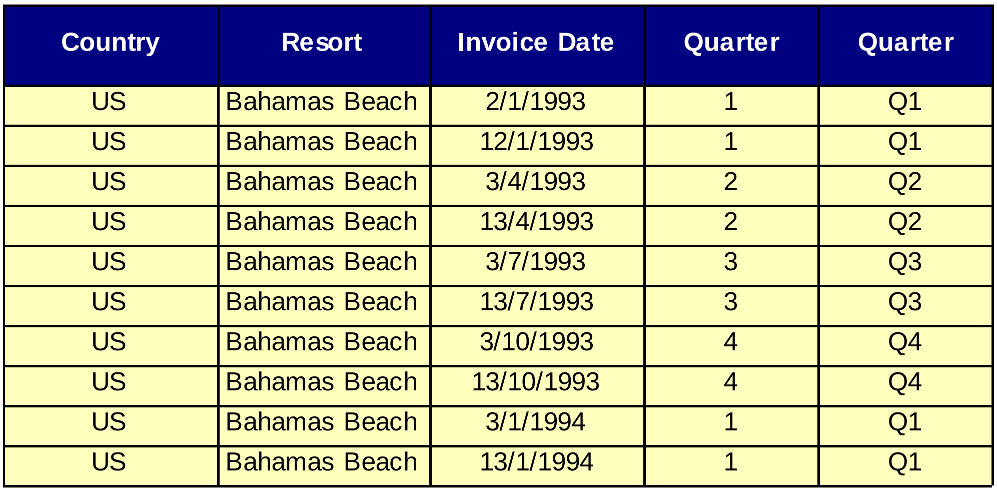
Begin Month	End Month	Months Between
12/3/1997	12/1/1997	2
15/7/1997	15/6/1997	1
19/8/1997	19/6/1997	2
20/9/1997	20/2/1997	7

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

4.13 Quarter

The Quarter Function returns the number of the quarter in the date. =Quarter(<date >)

For example: =Quarter(“05/08/1996”) returns 2 as the date falls in the second quarter of the year.



Country	Resort	Invoice Date	Quarter	Quarter
US	Bahamas Beach	2/1/1993	1	Q1
US	Bahamas Beach	12/1/1993	1	Q1
US	Bahamas Beach	3/4/1993	2	Q2
US	Bahamas Beach	13/4/1993	2	Q2
US	Bahamas Beach	3/7/1993	3	Q3
US	Bahamas Beach	13/7/1993	3	Q3
US	Bahamas Beach	3/10/1993	4	Q4
US	Bahamas Beach	13/10/1993	4	Q4
US	Bahamas Beach	3/1/1994	1	Q1
US	Bahamas Beach	13/1/1994	1	Q1

Text can also be added in front of the Quarter function
For example:

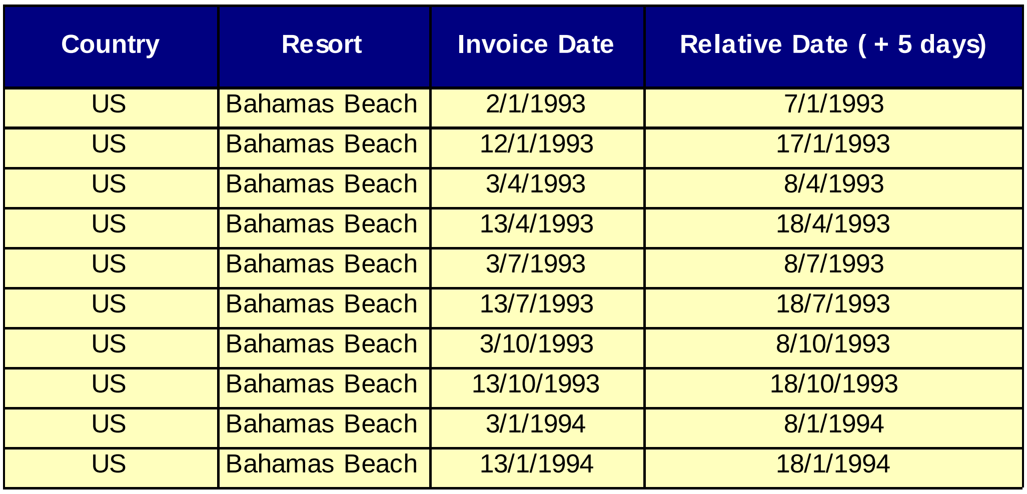
="Q" +Quarter(<Invoice Date>

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

4.14 RelativeDate

The RelativeDate Function returns the date that is equal to the specified date plus the specified number of days.
=RelativeDate(date, number of days)

For example: =RelativeDate('10-04-94', 5) returns 10-09-94.



Country	Resort	Invoice Date	Relative Date (+ 5 days)
US	Bahamas Beach	2/1/1993	7/1/1993
US	Bahamas Beach	12/1/1993	17/1/1993
US	Bahamas Beach	3/4/1993	8/4/1993
US	Bahamas Beach	13/4/1993	18/4/1993
US	Bahamas Beach	3/7/1993	8/7/1993
US	Bahamas Beach	13/7/1993	18/7/1993
US	Bahamas Beach	3/10/1993	8/10/1993
US	Bahamas Beach	13/10/1993	18/10/1993
US	Bahamas Beach	3/1/1994	8/1/1994
US	Bahamas Beach	13/1/1994	18/1/1994

Please note that the date format depends on the Regional Settings Short Date Format from the Regional Settings (in Settings/Control Panel/Regional Settings/Date Tab).
The format can also be set to a different format in the Number tab of the Cell Format dialog box.
Alternatively you can use the FormatDate Function to change the standard format to the required string format.

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

4.15 ToDate

The ToDate Function converts a character string into a date.
=ToDate(character string, string format)

For example: =ToDate(“19980112”,”yyyyddmm”)
Returns :01/12/98

Character 1	ToDate on Character 1	Character 2	ToDate on Character 2
199512	1/12/1995	12-Apr-98	12/4/1998
199611	1/11/1996	21-Apr-98	21/4/1998
199912	1/12/1999	23-Mar-98	23/3/1998

In order to convert a character string to a date, the date format set in the function needs to comply with the format specified by the character. With a character string: 12-apr-98 , the ToDate formula will be : =ToDate(“12-apr-98”,”dd-mmm-yy”)
With a character string: 199512 the ToDate formula needs to be: =ToDate(“199512”,”yyyymm”)
If the date format in the ToDate formula does not comply with the specified format of the character string this will produce a #ERROR in the report cells. (Note that with the ToDate any changes in the format will produce the error, so also when you are using a ‘/’ instead of a ‘-‘)

Please note that the format that the ToDate Formula returns is the date format that is set in the Regional Settings, so if the date format in the date tab in the Regional Settings is set to: dd/mm/yyyy , then the formula: =ToDate(“12-apr-98”,”dd-mmm-yy”) will return the value : “12/04/1998”
If it is set to “mm-dd-yy” the above formula will return the result of: “04-12-98”

In order to find out what character format was used in the Designer (as some users may not have rights to go to the designer) you need to check the ‘show list of values’ in the edit formula. This will come up when you highlight the object you are using in the formula. When this shows ‘12-apr-97’ you will need to use ‘dd-mmm-yy’ in the To Date Formula.

If it is required to change the date format that is returned, this can either be changed in the Regional Settings, or this can also be set to a different format in the Number tab of the Cell Format dialog box. Alternatively you can use the FormatDate function to change the standard format to the required format.

The Function ToDate is also important when you have run a setup with more than one language, for example French and English. If you are swapping between languages, then this may cause an error with the dates (or number formats). The reason being is that if you have run the report in English, you would have set the format to English date settings of dd/mm/yyyy and the French setting would need to be jj/mm/aaaa. When running this with BOLANGUAGE on EN then this will show the correct value, but when you change the BOLANGUAGE to FR this will give a #ERROR in the report. (Note that this BOLANGUAGE changes when you go to Tools / Options and change the Language you wish to run BO with from for example English to French – the BOLANGUAGE value will become FR.)

To solve this, you can make use of the ToDate Function in conjunction with the Function ApplicationValue.

The formula to use is:

```
=  If ApplicationValue("BOLANGUAGE") ="FR" Then
ToDate(<character> , "jj/mm/aa") Else ToDate(<character>
,"dd/mm/yy")
```

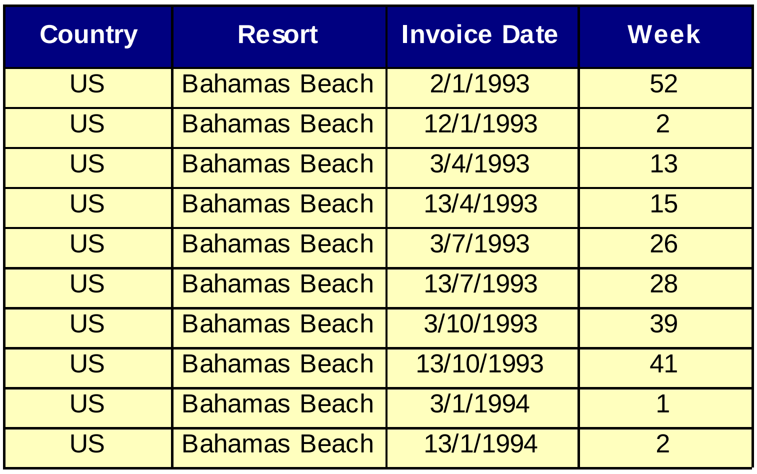
This will set the value correct, based on the language settings in the BOLANGUAGE file, and the date format in the Regional Settings.

For more information on ApplicationValue see Misc Functions

4.16 Week

The Week Function returns the number of the week of a date. The numbers are 1 to 52 or 53, depending on the year.
=Week(date)
Please note that Weeks begin on Monday, starting with the first full week in the year.
Thus: Week 1 of 1997 is Monday, January 6 through Sunday, January 12
Week 2 of 1997 is Monday, January 13 through Sunday, January 19
...
Week 52 of 1997 is Monday, December 29, 1997 through Sunday, January 4, 1998
Week 1 of 1998 is Monday, January 5 through Sunday, January 11.

For example: =Week("08/05/1995") returns 31



Country	Resort	Invoice Date	Week
US	Bahamas Beach	2/1/1993	52
US	Bahamas Beach	12/1/1993	2
US	Bahamas Beach	3/4/1993	13
US	Bahamas Beach	13/4/1993	15
US	Bahamas Beach	3/7/1993	26
US	Bahamas Beach	13/7/1993	28
US	Bahamas Beach	3/10/1993	39
US	Bahamas Beach	13/10/1993	41
US	Bahamas Beach	3/1/1994	1
US	Bahamas Beach	13/1/1994	2

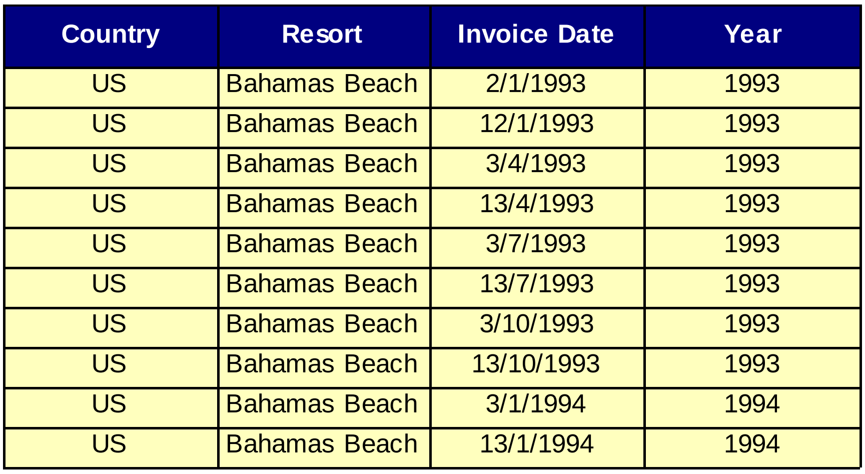
The number returned is standard formatted with two decimal places, but can be reformatted in the Number tab in the Format Cell menu.

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

4.17 Year

The Year Function returns the year of the date.
=Year(date)

For example: If Reservation Date is a date-type variable,
=Year(“08/05/1995”) returns 1995



Country	Resort	Invoice Date	Year
US	Bahamas Beach	2/1/1993	1993
US	Bahamas Beach	12/1/1993	1993
US	Bahamas Beach	3/4/1993	1993
US	Bahamas Beach	13/4/1993	1993
US	Bahamas Beach	3/7/1993	1993
US	Bahamas Beach	13/7/1993	1993
US	Bahamas Beach	3/10/1993	1993
US	Bahamas Beach	13/10/1993	1993
US	Bahamas Beach	3/1/1994	1994
US	Bahamas Beach	13/1/1994	1994

As with most Date Functions this can be set on a date object (see report), date function (for example CurrentDate()) or on single date values.

5 Logical Functions

The Logical Functions are divided into the following Functions:

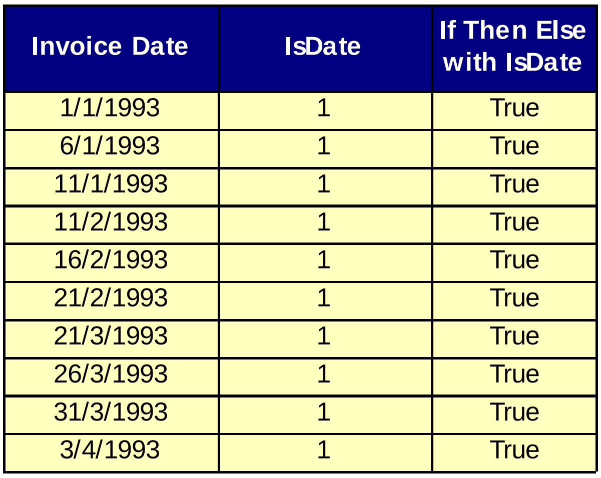
IsDate
IsError
IsLogical
IsNull
IsNumber
IsString
IsTime

This Chapter will now have a close look at the individual Logical Functions

5.1 IsDate

The IsDate Function checks whether a constant or a variable is of "date" type. Used with If Then Else. IsDate(character string or number or date)

For example:If Reservation Date is a date-type variable, =IsDate(<Reservation Date>) Then "true" Else "false" returns "true".



Invoice Date	IsDate	If Then Else with IsDate
1/1/1993	1	True
6/1/1993	1	True
11/1/1993	1	True
11/2/1993	1	True
16/2/1993	1	True
21/2/1993	1	True
21/3/1993	1	True
26/3/1993	1	True
31/3/1993	1	True
3/4/1993	1	True

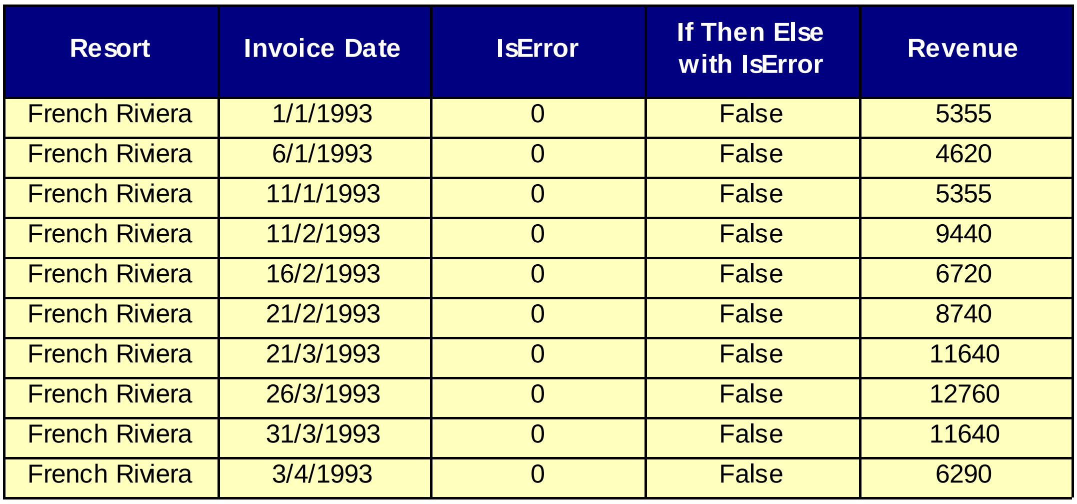
Note that you can use this function without using If Then Else, e.g., =IsDate(<Reservation Date>). In this case, the values returned are 1 (true), or 0 (false). You can format these values in the Number tab of the Format Cell dialog box.

This Function can be used with objects, as seen in the report, or with single values, trying to find out if they are of date-type or not

5.2 IsError

The IsError Function checks whether a constant or a variable returns an error. =IsError (character string or number or date) This function can be used in combination with an If Then Else Statement or without. In this case, the values returned are 0 (false) or 1 (true). You can format these values by using the Number tab of the Format Cell dialog box.

For example: =If IsError(<date>) Then "True" Else "False", returning False if there is no error.



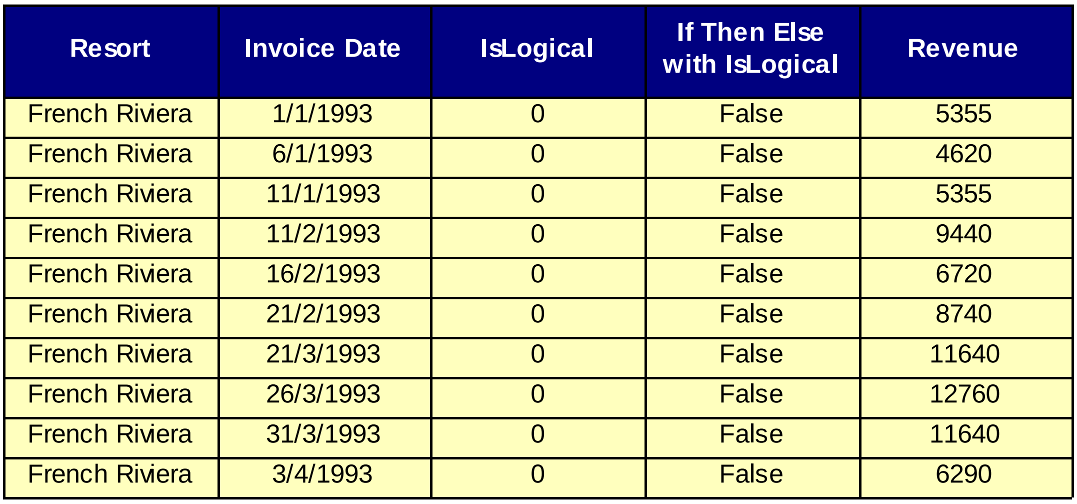
Resort	Invoice Date	IsError	If Then Else with IsError	Revenue
French Riviera	1/1/1993	0	False	5355
French Riviera	6/1/1993	0	False	4620
French Riviera	11/1/1993	0	False	5355
French Riviera	11/2/1993	0	False	9440
French Riviera	16/2/1993	0	False	6720
French Riviera	21/2/1993	0	False	8740
French Riviera	21/3/1993	0	False	11640
French Riviera	26/3/1993	0	False	12760
French Riviera	31/3/1993	0	False	11640
French Riviera	3/4/1993	0	False	6290

This Function can be used with objects, as seen in the report, or with single values, trying to find out if they contain an error or not

5.3 IsLogical

The IsLogical Function checks whether a variable or constant is a logical. =IsLogical(character string or number or date)
This function can be used in combination with an If Then Else Statement or without. In this case, the values returned are 0 (false) or 1 (true). You can format these values by using the Number tab of the Format Cell dialog box.

For example: =If IsLogical(<MyLogicalVariable>) Then "true" Else "false" is true, where MyLogicalVariable equal to Revenue>10000.



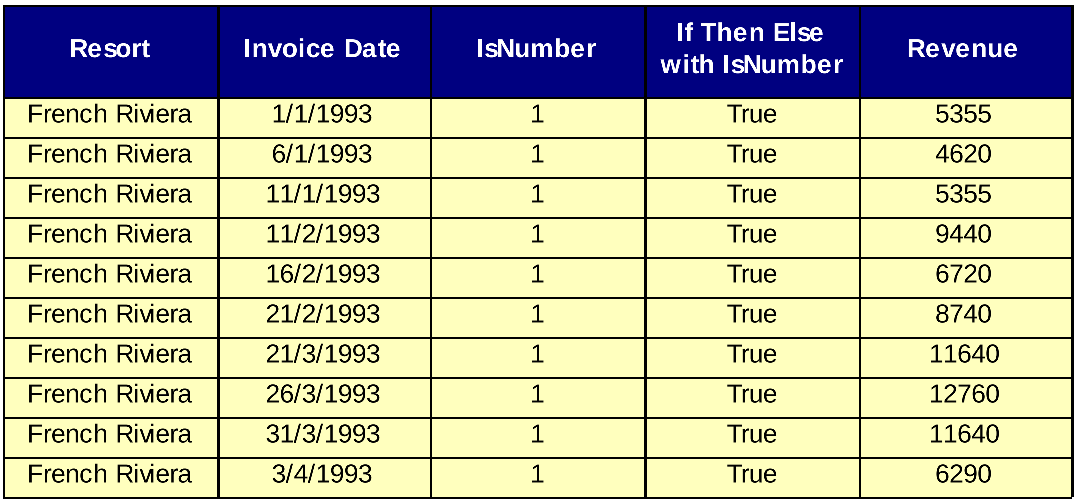
Resort	Invoice Date	IsLogical	If Then Else with IsLogical	Revenue
French Riviera	1/1/1993	0	False	5355
French Riviera	6/1/1993	0	False	4620
French Riviera	11/1/1993	0	False	5355
French Riviera	11/2/1993	0	False	9440
French Riviera	16/2/1993	0	False	6720
French Riviera	21/2/1993	0	False	8740
French Riviera	21/3/1993	0	False	11640
French Riviera	26/3/1993	0	False	12760
French Riviera	31/3/1993	0	False	11640
French Riviera	3/4/1993	0	False	6290

This Function can be used with objects, as seen in the report, or with single values, trying to find out if they contain an error or not

5.4 IsNumber

The IsNumber Function checks whether a variable or constant is a number. =IsNumber(character string or number or date) This function can be used in combination with an If Then Else Statement or without. In this case, the values returned are 0 (false) or 1 (true). You can format these values by using the Number tab of the Format Cell dialog box.

For example: =If IsNumber(<Revenue>) Then "true" Else "false" is true.



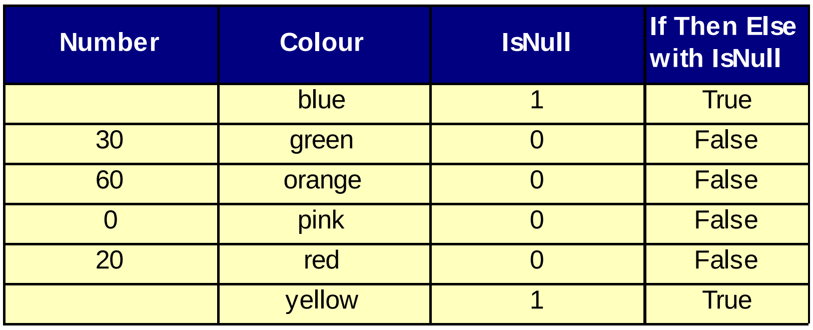
Resort	Invoice Date	IsNumber	If Then Else with IsNumber	Revenue
French Riviera	1/1/1993	1	True	5355
French Riviera	6/1/1993	1	True	4620
French Riviera	11/1/1993	1	True	5355
French Riviera	11/2/1993	1	True	9440
French Riviera	16/2/1993	1	True	6720
French Riviera	21/2/1993	1	True	8740
French Riviera	21/3/1993	1	True	11640
French Riviera	26/3/1993	1	True	12760
French Riviera	31/3/1993	1	True	11640
French Riviera	3/4/1993	1	True	6290

This Function can be used with objects, as seen in the report, or with single values, trying to find out if they are of numeric type or not.

5.5 IsNull

The IsNull Function checks whether a constant or variable is null (meaning #Empty). =IsNull(character string or number or date)
This function can be used in combination with an If Then Else Statement or without. In this case, the values returned are 0 (false) or 1 (true). You can format these values by using the Number tab of the Format Cell dialog box.

For example: =If IsNull(<Value> Then "true" Else "false" is false.



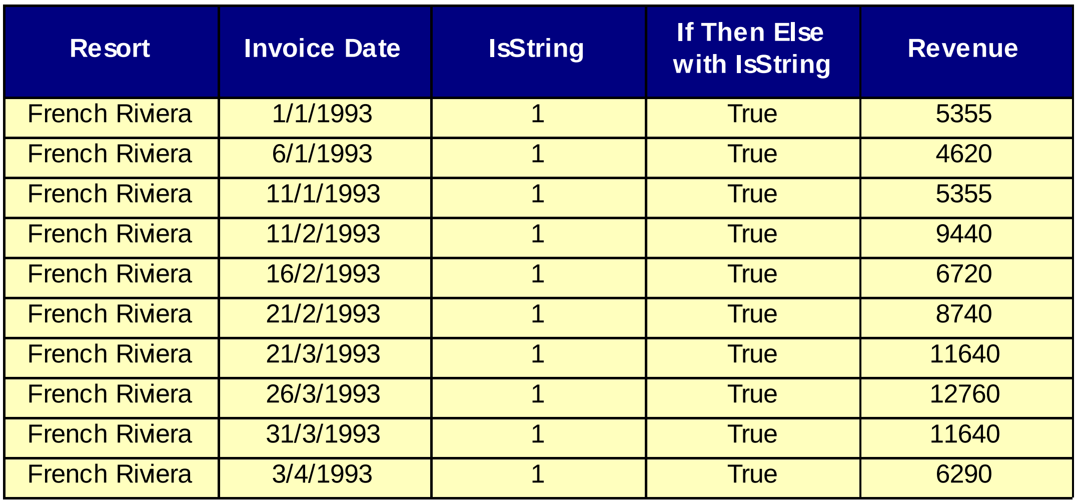
Number	Colour	IsNull	If Then Else with IsNull
	blue	1	True
30	green	0	False
60	orange	0	False
0	pink	0	False
20	red	0	False
	yellow	1	True

This Function can be used with objects, as seen in the report, or with single values, trying to find out if they contain null or #Empty values

5.6 IsString

The IsString Function checks whether a variable or constant is a character string. =IsString(character string or number or date) This function can be used in combination with an If Then Else Statement or without. In this case, the values returned are 0 (false) or 1 (true). You can format these values by using the Number tab of the Format Cell dialog box.

For example: =If IsString(<First Name>) Then "true" Else "false" is true.



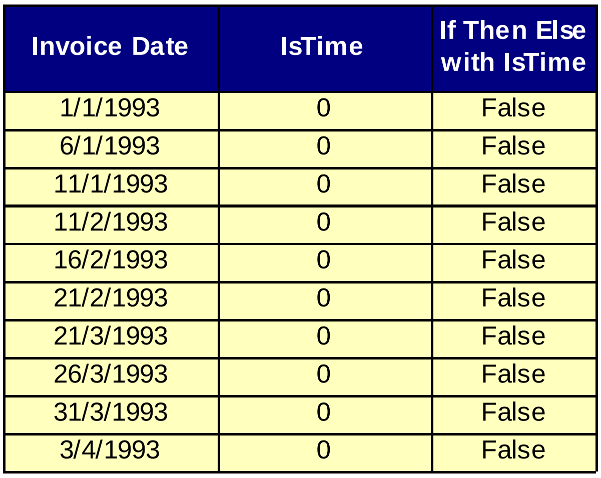
Resort	Invoice Date	IsString	If Then Else with IsString	Revenue
French Riviera	1/1/1993	1	True	5355
French Riviera	6/1/1993	1	True	4620
French Riviera	11/1/1993	1	True	5355
French Riviera	11/2/1993	1	True	9440
French Riviera	16/2/1993	1	True	6720
French Riviera	21/2/1993	1	True	8740
French Riviera	21/3/1993	1	True	11640
French Riviera	26/3/1993	1	True	12760
French Riviera	31/3/1993	1	True	11640
French Riviera	3/4/1993	1	True	6290

This Function can be used with objects, as seen in the report, or with single values, trying to find out if they are of character-type or not.

5.7 IsTime

The IsTime Function checks whether a variable or constant is a time. =IsTime(character string or number or date)
This function can be used in combination with an If Then Else Statement or without. In this case, the values returned are 0 (false) or 1 (true). You can format these values by using the Number tab of the Format Cell dialog box.

For example: =IsTime(<ReportTime>) Then 0 Else 1 is 0



Invoice Date	IsTime	If Then Else with IsTime
1/1/1993	0	False
6/1/1993	0	False
11/1/1993	0	False
11/2/1993	0	False
16/2/1993	0	False
21/2/1993	0	False
21/3/1993	0	False
26/3/1993	0	False
31/3/1993	0	False
3/4/1993	0	False

This Function can be used with objects, as seen in the report, or with single values, trying to find out if they are of character-type or not.

6 Document Functions

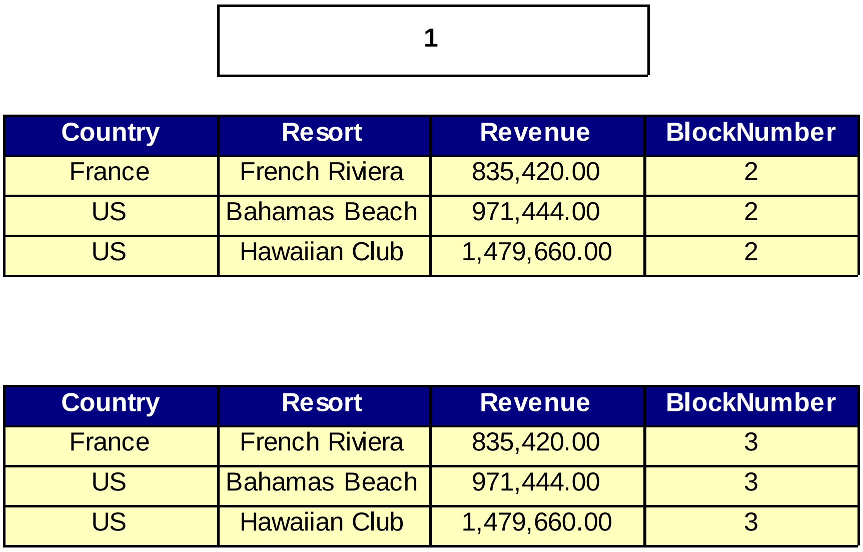
The Document Functions are the following:

BlockNumber
ColumnNumber
DocumentAuthor
DocumentDate
DocumentName
DocumentPartiallyRefreshed
DocumentTime
LastPrintDate
LineNumber
NumberOfPages
Page
SectionNumber

This Chapter will now concentrate on the individual Document Functions

6.1 BlockNumber

The BlockNumber() Function will return the number of the current block, starting from 1.



1

Country	Resort	Revenue	BlockNumber
France	French Riviera	835,420.00	2
US	Bahamas Beach	971,444.00	2
US	Hawaiian Club	1,479,660.00	2

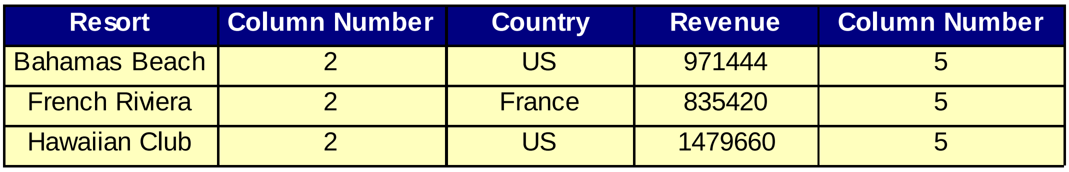
Country	Resort	Revenue	BlockNumber
France	French Riviera	835,420.00	3
US	Bahamas Beach	971,444.00	3
US	Hawaiian Club	1,479,660.00	3

Please note that the Title of the report is also classed as a Block and will be classed as BlockNumber 1.
All new reports will get a BlockNumber accordingly.

Please note that the number returned can be formatted with decimal places in the number tab of the Format Cell menu.

6.2 ColumnNumber

The ColumnNumber() Function returns the number of the current column, starting from 1.



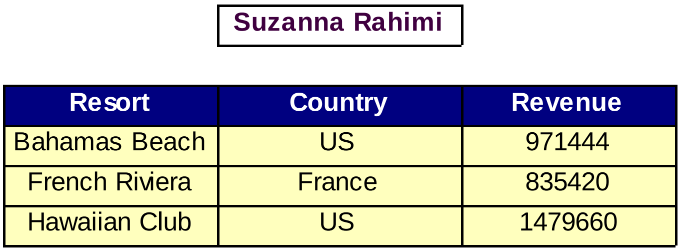
Resort	Column Number	Country	Revenue	Column Number
Bahamas Beach	2	US	971444	5
French Riviera	2	France	835420	5
Hawaiian Club	2	US	1479660	5

The ColumnNumber Function in the report returns 2, whereas the column is the second column in the report, and the fourth column will return the number 4 to indicate the fourth column.

Please note that the number returned can be formatted with decimal places in the number tab of the Format Cell menu.

6.3 DocumentAuthor

The DocumentAuthor() Function will return the name of the user that has created the document.

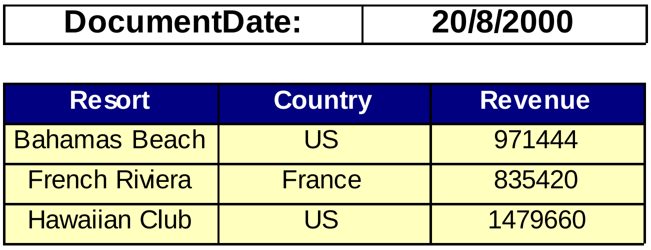


Suzanna Rahimi

Resort	Country	Revenue
Bahamas Beach	US	971444
French Riviera	France	835420
Hawaiian Club	US	1479660

6.4 DocumentDate

The DocumentDate () Function returns the date on which the report has last been saved. This can be checked in the directory where the document is stored (UserDocs), as the last modification date will comply with this date.



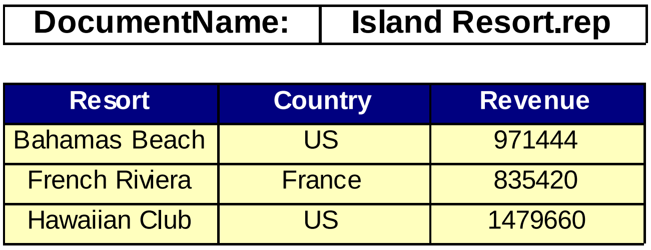
DocumentDate:	20/8/2000

Resort	Country	Revenue
Bahamas Beach	US	971444
French Riviera	France	835420
Hawaiian Club	US	1479660

The date format returned is based on the Regional Settings in the Control Panel. It can also be reformatted in the Number tab in the Format Cell menu.

6.5 DocumentName

The DocumentName() Function returns the name of the document.



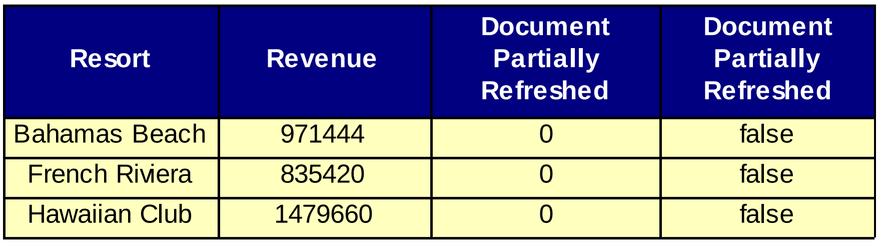
DocumentName:	**Island Resort.rep**

Resort	Country	Revenue
Bahamas Beach	US	971444
French Riviera	France	835420
Hawaiian Club	US	1479660

6.6 DocumentPartiallyRefreshed

The DocumentPartiallyRefreshed() Function returns a Boolean value indicating whether the results are partial or not. The value TRUE means that results are partial.
This function can be used in combination with an If Then Else Statement or without. In this case, the values returned are 0 (false) or 1 (true).

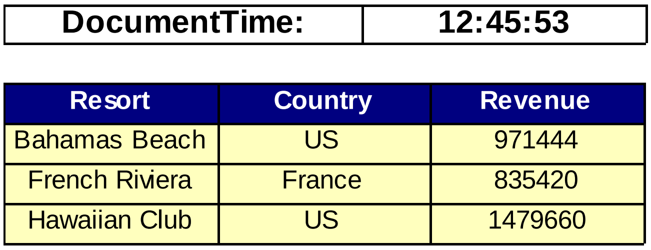
You can format these values by using the Number tab of the Format Cell dialog box.



Resort	Revenue	Document Partially Refreshed	Document Partially Refreshed
Bahamas Beach	971444	0	false
French Riviera	835420	0	false
Hawaiian Club	1479660	0	false

6.7 DocumentTime

The DocumentTime() Function returns the time on which the document was last saved . This can be checked in the directory where the document is stored, as it will comply with the last modification time.



DocumentTime:	12:45:53

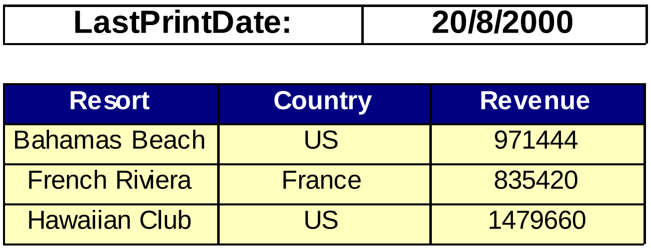
Resort	Country	Revenue
Bahamas Beach	US	971444
French Riviera	France	835420
Hawaiian Club	US	1479660

The time format that is returned is based on the Regional Settings format in the Control Panel.

Please note that this can not be altered in the Format Cell menu.

6.8 LastPrintDate

The LastPrintDate() Function returns the date that the report was last printed.



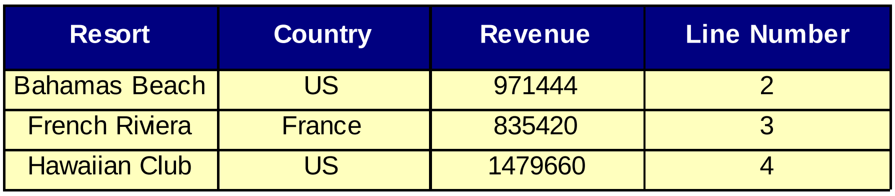
LastPrintDate:	**20/8/2000**

Resort	Country	Revenue
Bahamas Beach	US	971444
French Riviera	France	835420
Hawaiian Club	US	1479660

The date format returned is based on the Regional Settings in the Control Panel. It can also be reformatted in the Date option in the Format Cell menu.

6.9 LineNumber

The LineNumber() Function returns the number of the current line, starting from 1.
All rows in the block are counted, including header rows and footer rows (including the calculation rows), but only the body of the report will show the line numbers displayed.



Resort	Country	Revenue	Line Number
Bahamas Beach	US	971444	2
French Riviera	France	835420	3
Hawaiian Club	US	1479660	4

Please note that the number returned can be formatted with decimal places in the number tab of the Format Cell menu.

6.10 NumberOfPages

The NumberOfPages() Function returns the number of pages that are in the report.

Please note that this goes in combination with the View Page Layout option in the Taskbar. If this is not set, then the number of pages function will return –1 (this is the same when the report has been saved as HTML, as the HTML report does not have a page layout function and displays the data as one page.)

The format the number of pages is returned as is standard with two decimal places. This can be altered in the number tab in the Format Cell menu.

*Tip: How to display the page number of the number of pages?
Like: page 1 /20 ,or page 1 of 20?

Formula:
=Page()+"/"+NumberOfPages()
Will return: 1/20
= "page"+ Page()+"of "+NumberOfPages()
Will return: page 1 of 20

6.11 Page

The Page() Function will return the page number of each page in the report.

Please note that this goes in combination with the View Page Layout option in the Taskbar. If this is not set, then the number of pages function will return 0 (this is the same (but shown as –1 rather than 0) when the report has been saved as HTML, as the HTML report does not have a page layout function and displays the data as one page.)

The format the number of pages is returned as is standard with two decimal places. This can be altered in the number tab in the Format Cell menu.

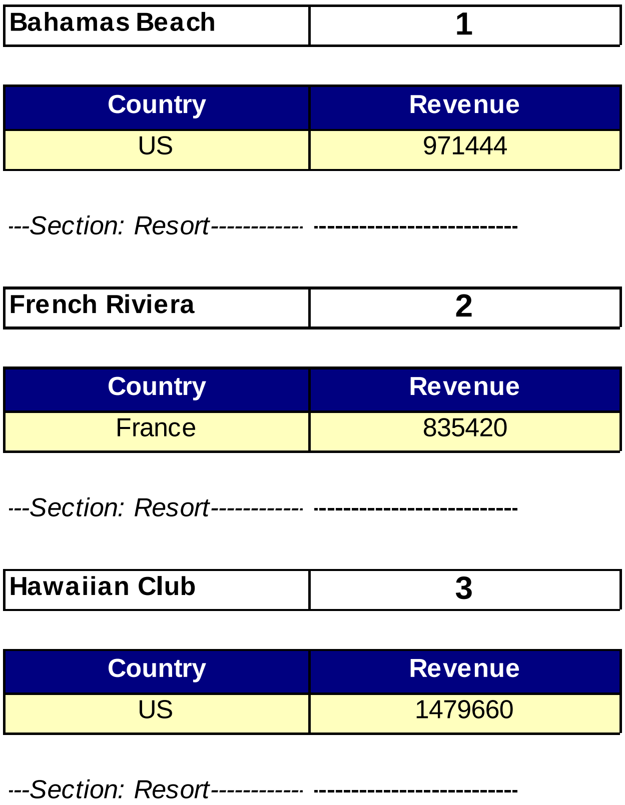
*Tip: How to display the page number of the number of pages?
Like: page 1 /20 ,or page 1 of 20?

Formula:
=Page()+"/"+NumberOfPages()
Will return: 1/20
= "page"+ Page()+"of "+NumberOfPages()
Will return: page 1 of 20

6.12 SectionNumber

The SectionNumber() Function returns the number of the current section, starting from 1.

Please note that the number returned is standard with two decimal places and can be formatted without decimal places in the number tab of the Format Cell menu. (This can also be set in the Regional Settings in Control Panel, but this will remove all decimal paces everywhere)



Bahamas Beach	**1**

Country	Revenue
US	971444

---Section: Resort----------- --------------------------

French Riviera	**2**

Country	Revenue
France	835420

---Section: Resort----------- --------------------------

Hawaiian Club	**3**

Country	Revenue
US	1479660

---Section: Resort----------- --------------------------

7 Data Provider Functions

The Data Provider Functions are divided into the following Functions:

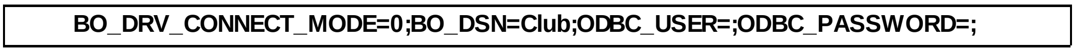
Connection
DataProvider
DataProviderSQL
DataProviderType
LastExecutionDate
LastExecutionTime
NumberOfDataProviders
NumberOfRows
UniverseName
UserResponse

This Chapter will now have a close look at the individual Data Provider Functions

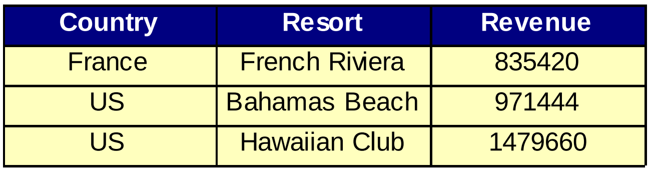
7.1 Connection

The Connection Function returns the parameters of the database connection used by the data provider to query the database. =Connection(Dataprovider)

In order to activate the formula you need to insert the Dataprovider name. This is in actual fact the name of the query. This can be found in the Data Manager in the Definion tab, under Name: “Query 1 with Beach”
The formula becomes: =Connection (“Query 1 with Beach”)



BO_DRV_CONNECT_MODE=0;BO_DSN=Club;ODBC_USER=;ODBC_PASSWORD=;

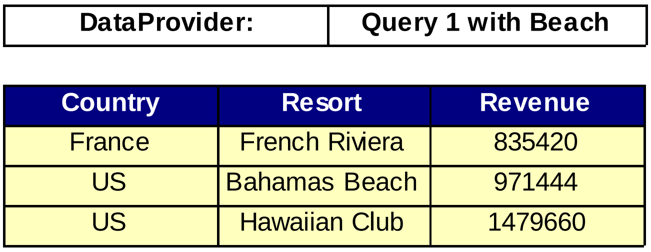


Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

7.2 DataProvider

The DataProvider Function returns the name of the data provider that retrieved the data for the variable in the block.
=DataProvider(<object>)

For example if the query has been run on data from the Island Resort /Beach universe, then the name of the data provider is "Query 1 with Beach". By selecting the formula: =DataProvider(<Resort>), the value returned is "Query 1 with Beach", which is the name of the Query.
(This can be found in the Data Manager in the Definition tab, under Name: Query 1 with Beach))

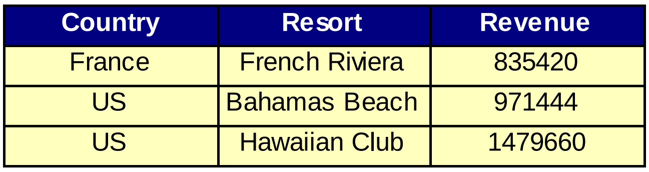


DataProvider:	**Query 1 with Beach**

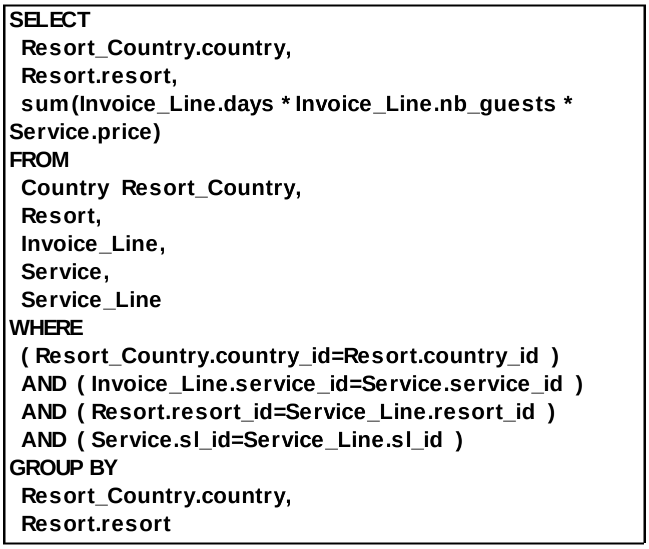
Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

7.3 DataProviderSQL

The DataProviderSQL Function Returns the SQL of the data provider, or an error if no SQL script is generated (in the case of a personal data file from Microsoft Excel, for example).
=DataProviderSQL(DataProvider)



Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660



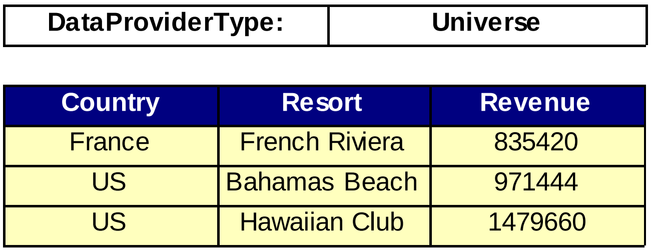
```
SELECT
  Resort_Country.country,
  Resort.resort,
  sum(Invoice_Line.days * Invoice_Line.nb_guests *
Service.price)
FROM
  Country  Resort_Country,
  Resort,
  Invoice_Line,
  Service,
  Service_Line
WHERE
  ( Resort_Country.country_id=Resort.country_id  )
  AND  ( Invoice_Line.service_id=Service.service_id  )
  AND  ( Resort.resort_id=Service_Line.resort_id  )
  AND  ( Service.sl_id=Service_Line.sl_id  )
GROUP BY
  Resort_Country.country,
  Resort.resort
```

In order to activate the formula you need to insert the Dataprovider name. This is in actual fact the name of the query. This can be found in the Data Manager in the Definion tab, under Name: “Query 1 with Beach”
The formula becomes:
=DataProviderSQL (“Query 1 with Beach”)

In actual fact, this function will display the SQL script as shown in the SQL button in the Query Panel

7.4 DataProviderType

The DataProviderType returns the type of the data provider. The values that can be returned are "Universe" (for queries), "Stored procedures", "Free-hand SQL" and "Personal data". Depending on what option has been used in the Welcome Wizard of Business Objects to run a query against , this will display the type. The formula is based on :
DataProviderType(DataProvider)
In order to activate the formula you need to insert the Dataprovider name, which in actual fact is the name of the query, and can be found in the Data Manager in the Definion tab, under Name: “Query 1 with Beach”
The formula becomes:
=DataProviderType (“Query 1 with Beach”) returning “Universe”

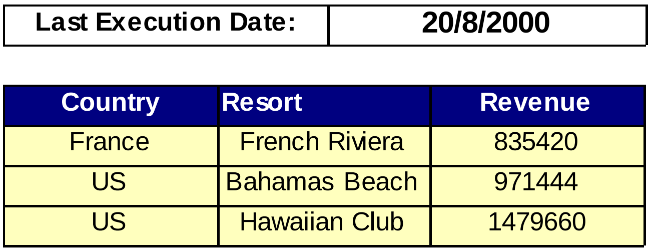


DataProviderType:	**Universe**

Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

7.5 LastExecutionDate

The LastExecutionDate Function returns the date that the data provider was last run in the Query Panel or refreshed in the Data Manager. This date will be the date that is also displayed in the Data Manager in the Definition tab under date.
Formula: =LastExecutionDate(DataProvider)
In order to activate the formula you need to insert the Dataprovider name, which in actual fact is the name of the query, and can be found in the Data Manager in the Definion tab, under Name: “Query 1 with Beach”
The formula becomes:
=LastExecutionTime(“Query 1 with Beach”)returning the date on which the report has last been run or refreshed.



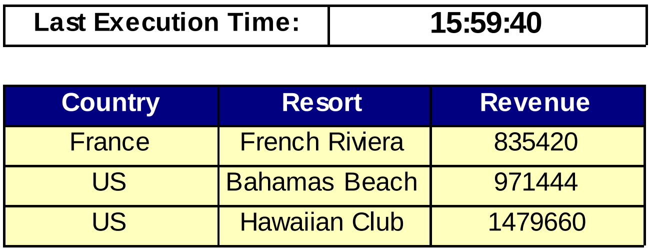
Last Execution Date:	**20/8/2000**

Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

The date format is based on the date format in the Regional Settings, but can also be adapted on reporter level via the Number tab and the date option in the Format Cell menu.

7.6 LastExecutionTime

The LastExecutionTime Function returns the time that the data provider was last run in the Query Panel or refreshed in the Data Manager. This time will be the time that is also displayed in the Data Manager in the Definition tab under time.
Formula: =LastExecutionTime(DataProvider)
In order to activate the formula you need to insert the Dataprovider name, which in actual fact is the name of the query, and can be found in the Data Manager in the Definion tab, under Name: “Query 1 with Beach”
The formula becomes:
=LastExecutionTime(“Query 1 with Beach”)returning the time on which the report has last been run or refreshed.



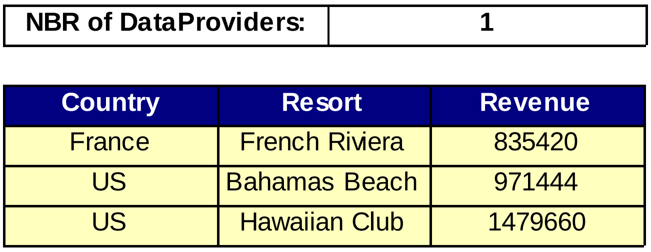
Last Execution Time:	**15:59:40**

Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

The time format is based on the date format in the Regional Settings, and can not be adapted on reporter level via the Number tab and the date option in the Format Cell menu.

7.7 NumberOfDataProviders

The NumberOfDataProviders() returns the number of data providers in the current document. There may be a number of queries in the document, and this function will indicate if the queries come from one or more data providers.



NBR of DataProviders:	**1**

Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

The number can be altered in the Number tab in the Format Cell menu.

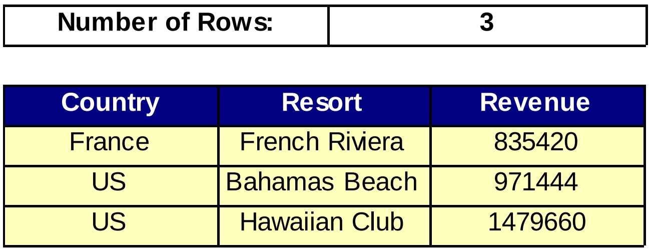
Tip: This function can be set together with other functions, for example:
=NumberOfDataProviders()+"/"+DataProviderType("Query 1 with Beach")
This will return: 1/Universe

7.8 NumberOfRows

The NumberOfRows Function returns the number of rows retrieved by the specified data provider.
Formula: =NumberOfRows(DataProvider)

In order to activate the formula you need to insert the Dataprovider name, which in actual fact is the name of the query, and can be found in the Data Manager in the Definion tab, under Name: “Query 1 with Beach”
The formula becomes:
=NumberOfRows(“Query 1 with Beach”)returning number of rows of data that is stored in the DataManager.

Please note that if a report contains a filter, or if an object is removed from the query, resulting in less rows in the query, this will not change the number of rows retrieved from the DataProvider. The number of rows retrieved is the unfiltered number.



Number of Rows:	**3**

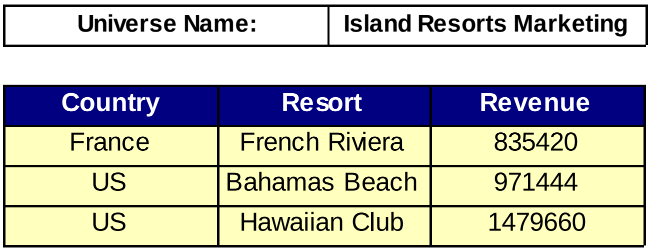
Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

The number can be altered in the Number tab in the Format Cell menu.

7.9 UniverseName

The UniverseName Function returns the name of the active universe used by the data provider.
The Formula is: =UniverseName(DataProvider)

In order to activate the formula you need to insert the Dataprovider name, which in actual fact is the name of the query, and can be found in the Data Manager in the Definion tab, under Name: “Query 1 with Beach”
The formula becomes:
=UniverseName(“Query 1 with Beach”)returning the long universe name .

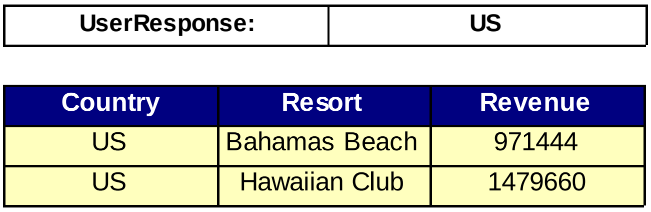


Universe Name:	**Island Resorts Marketing**

Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

7.10 UserResponse

The UserResponse Function returns the response to the specified prompt. This prompt is written when the data provider (query or free-hand SQL script) is set up. It appears when you run the query or SQL script. The response to the prompt is the value returned by the UserResponse function.
This can be used for prompts set in the designer or in the query panel.
The Formula is: =UserResponse(DataProvider, prompt)
For example: [Which resort?] is a prompt for the query named “Query 1 with Beach”. The response to the prompt is "Bahamas Beach". Therefore, the formula =UserResponse("Query 1 with Beach", "Which resort?") returns "Bahamas Beach".



UserResponse:	US

Country	Resort	Revenue
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

In the report the country has been set as a prompt:
@Prompt('type country','A','Resort\Country',multi,free)
The user has selected US as the country.
The UserResponse will be as follows:
=UserResponse(“Query 1 with Beach”, “type country”)
Returning US, as the user has entered this as the prompt value.

The same would be when the prompt is set in the query panel:
Country Equal To EnterPrompt “enter country”
The UserResponse would be:
=UserResponse(“Query 1 with Beach”,”enter country”)
Returning US

8 Misc Functions

The Miscellaneous Functions are the following:

ApplicationValue
CurrentUser
GetProfileNumber
GetProfileString
NameOf
NoFilter
Previous
RowIndex

This Chapter will now have a close look at the individual Miscellaneous Functions

8.1 ApplicationValue

The ApplicationValue Function retrieves the user variable that was set in a script. =ApplicationValue("ScriptVariable")

For example:

A user script contains the lines:
set myvar1="US"
set myvar2="FR"

The variable can be used in building a complex filter, with the current value taken when referenced with the ApplicationValue function, as in:

=ApplicationValue("myvar1") returning US

Note that the script variable can be defined in a text file and referenced with the -vars option on the BusinessObjects command line, for example:

businessobjects.exe -vars filename

where filename contains one or more variable assignments:

name of variable
content of variable

The function can also be used to find out the language setting of business objects :

= ApplicationValue("BOLANGUAGE") returns EN if the language is set to English

Note that this function can be used in combination with an IF THEN ELSE statement.
For example: If ApplicationValue("BOLANGUAGE")=”EN”Then “UK” Else If ApplicationValue("BOLANGUAGE")=”FR” Then “France”

The ApplicationValue is also important when you have run a setup with more than one language, for example French and English. If you are swapping between languages, then this may cause an error with the dates (or number formats). The reason being is that if you have run the report in English, you would have set the format to English date settings of dd/mm/yyyy and the French setting would need to be jj/mm/aaaa. When running this with BOLANGUAGE on EN then this will show the correct value, but when you change the BOLANGUAGE to FR this will give a #ERROR in the report. (Note that this BOLANGUAGE changes when you go to Tools / Options and change the Language you wish to run BO with from for example English to French – the BOLANGUAGE value will become FR.)

To solve this, you can make use of the ApplicationValue Function in conjunction with the Function FormatDate.

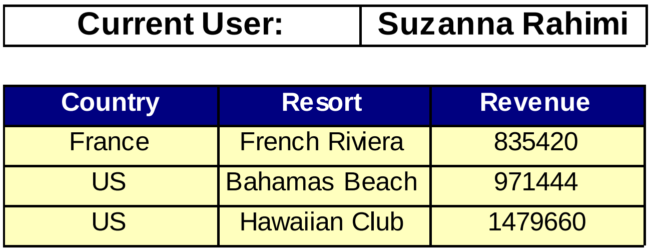
The formula to use is:

```
=  If ApplicationValue("BOLANGUAGE") ="FR" Then
FormatDate(<date> , "jj/mm/aa") Else FormatDate(<date>
,"dd/mm/yy")
```

This will set the value correct, based on the language settings in the BOLANGUAGE file. This will be the same case with the ToDate function.

8.2 CurrentUser

The CurrentUser() Function returns the name of the user who is working on the document. The name is the user's BusinessObjects user name, used when launching the application.



Current User:	Suzanna Rahimi

Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

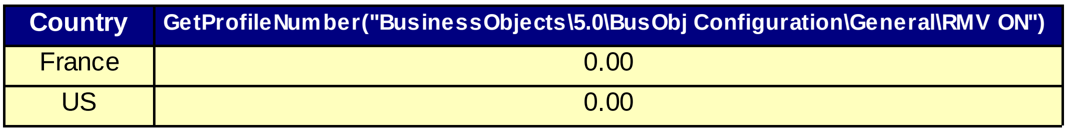
8.3 GetProfileNumber

The GetProfileNumber() function can be used to retrieve registry values included in the BusinessObjects key.
They directly include the registry path
"HKEY_LOCAL_MACHINE\Software\BusinessObjects" .

For example:
=GetProfileNumber("BusObj Configuration\General\RMV ON")
returns the value "0.00".

NOTE: The above example is based on BusinessObjects v4. In Version 5 you have an extra BusinessObjects\5.0 path in the registry, so the actual string becomes:

=GetProfileNumber("BusinessObjects\5.0\BusObj Configuration\General\RMV ON")



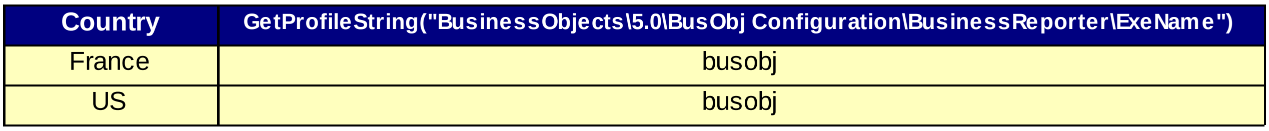
Country	GetProfileNumber("BusinessObjects\5.0\BusObj Configuration\General\RMV ON")
France	0.00
US	0.00

8.4 GetProfileString

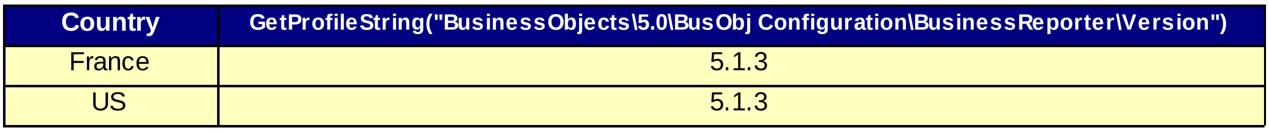
The GetProfileString() function can be used to retrieve registry values included in the BusinessObjects key.
They directly include the registry path "HKEY_LOCAL_MACHINE\Software\BusinessObjects" .

For examples:

=GetProfileString
("BusObj Configuration\BusinessReporter\ExeName")
returns the string "busobj".



Country	GetProfileString("BusinessObjects\5.0\BusObj Configuration\BusinessReporter\ExeName")
France	busobj
US	busobj

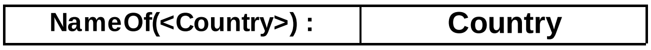


Country	GetProfileString("BusinessObjects\5.0\BusObj Configuration\BusinessReporter\Version")
France	5.1.3
US	5.1.3

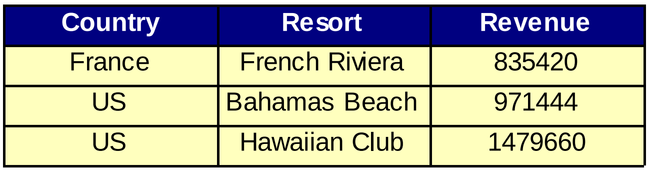
8.5 NameOf

The NameOf Function returns the name of the variable as a character string. =NameOf(<variable>)

For example: =NameOf(<Resort>) equals Resort.



NameOf(<Country>) :	Country



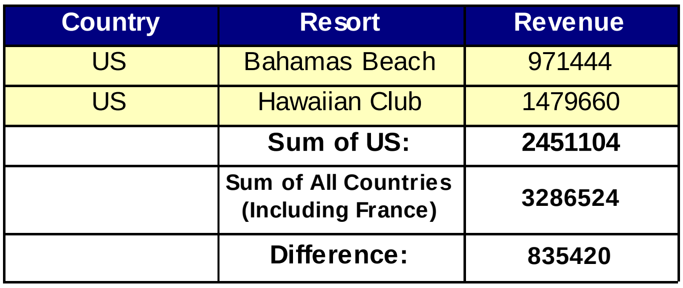
Country	Resort	Revenue
France	French Riviera	835420
US	Bahamas Beach	971444
US	Hawaiian Club	1479660

Note that BusinessObjects systematically uses the NameOf Function in column and row headers in reports.

8.6 NoFilter

The NoFilter Function returns the calculation performed on data with all filters on the data ignored. =NoFilter formula)

For example:
The report below displays a report with a filter placed on Country 'US'. The Revenue displays only the Revenue of the US, and the total Revenue of the US.
Applying the Formula:
=NoFilter(Sum(<Revenue>))
At the sum level, this will display the total Revenue for both France and US, as it will ignore the filter placed on Country.
Then you can calculate the difference of the two displayed totals (only in the report context)



Country	Resort	Revenue
US	Bahamas Beach	971444
US	Hawaiian Club	1479660
	Sum of US:	**2451104**
	Sum of All Countries (Including France)	**3286524**
	Difference:	**835420**

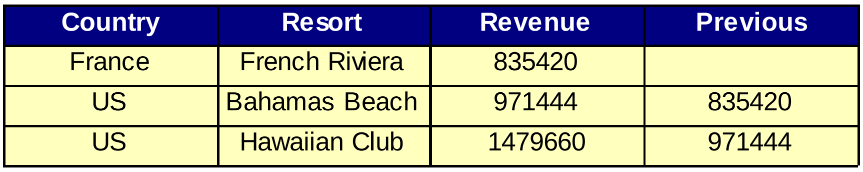
Another example: =NoFilter(Count(<Client>)) / Count (<Client>) returns the ratio of the number of total clients (unfiltered) to the number of clients with the current filter applied.

The NoFilter Function can also be used to obtain the complete total of a Ranked table. When for example a rank has been placed in the Slice & Dice menu for the top 3 customers, the sum will only give the total of the top 3. NoFilter around this sum will give the Grand Total of all customers.

Note: The use of the NoFilter Function does not alter the current filter.

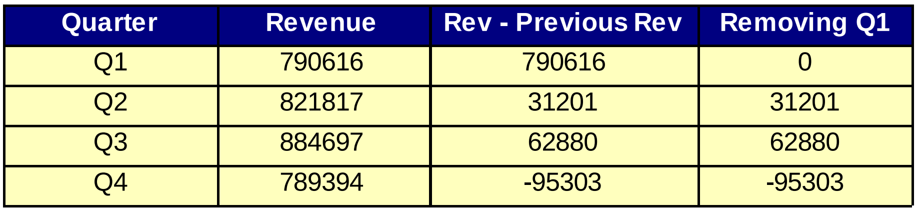
8.7 Previous

The Previous Function returns the previous value of the specified variable or object. =Previous (<object>)
This enables the user to compare values.
For example: =Previous(<Revenue>)



Country	Resort	Revenue	Previous
France	French Riviera	835420	
US	Bahamas Beach	971444	835420
US	Hawaiian Club	1479660	971444

The previous value is the next cell up in the column.
If you display revenue per quarter, then display revenue for *Qn -1* you can calculate the difference in revenue between the two quarters.



Quarter	Revenue	Rev - Previous Rev	Removing Q1
Q1	790616	790616	0
Q2	821817	31201	31201
Q3	884697	62880	62880
Q4	789394	-95303	-95303

See Column for Rev-Previous Rev.
The Previous Calculation will start from Q1 working upwards. As there is no value previous to the revenue in Q1 the calculation will be: 256860 – empty = 256860.
It will then look at the value for Q2 – Q1: Value 272490 – 256860 = 15630. Then Q3 - Q2: Value 288993 – 272490 = 16503. Then Q4 - Q3: Value 245211 - 288993= - 43782

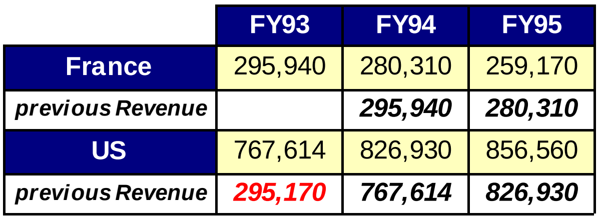
In the column called Removing Q1 value a formula is used with the Function IsNull in order to replace the value 256860 by 0. This shows that most functions van be used in conjunction with one and other.

Formula used:
= If IsNull(Previous(<Revenue>)) Then 0 Else <Revenue>-Previous(<Revenue>)

NOTE: all that is described above regarding the Previous functions counts for version 4.1.5 and above.
In all versions before 4.1.5, the Previous function did not behave as specified above.

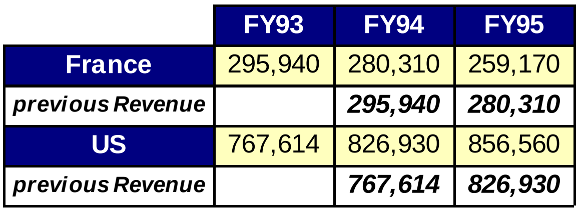
For example, if you have created a crosstab with Country, Year, and Revenue and Previous calculation objects for Revenue within an extra row (inserted between the country values), in versions before 4.15 the result is different:

Result up to v4.1.4:



	FY93	FY94	FY95
France	295,940	280,310	259,170
previous Revenue		***295,940***	***280,310***
US	767,614	826,930	856,560
previous Revenue	***295,170***	***767,614***	***826,930***

Result in 4.15 and versions:



	FY93	FY94	FY95
France	295,940	280,310	259,170
previous Revenue		***295,940***	***280,310***
US	767,614	826,930	856,560
previous Revenue		***767,614***	***826,930***

In a crosstab, it would display the previous data as if all the data were displayed in a list and did not break for each row. It can be said that the function looked at data listed in the microcube rather than at the previous cell in the report. The highlighted red value above is therefore incorrect, as it should be looking at the value of each country per previous year. This problem was fixed in 4.15

Tip: The Previous Function can also be used in order to calculate a Rolling Average. You can for example: calculate the Rolling Average over three Months.

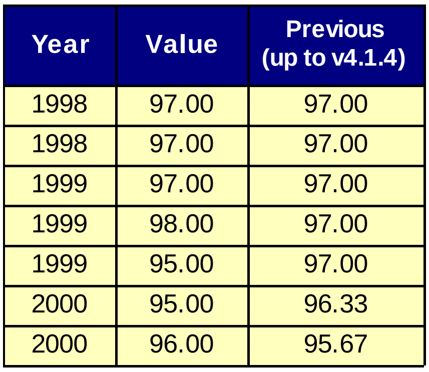
The Formula to use would be:

=(<Revenue>+Previous(<Revenue>)+Previous(Previous(<Revenue>)))/3

Due to the change in the behaviour of the Previous function since the release of BusinessObjects version 4.1.5 the results of the rolling average may differ.

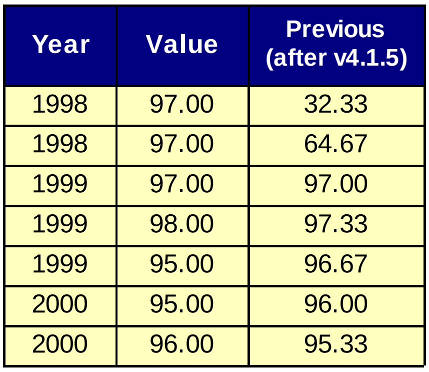
Results of Rolling Average:

Up to v4.1.4:



Year	Value	Previous (up to v4.1.4)
1998	97.00	97.00
1998	97.00	97.00
1999	97.00	97.00
1999	98.00	97.00
1999	95.00	97.00
2000	95.00	96.33
2000	96.00	95.67

After v4.1.5:



Year	Value	Previous (after v4.1.5)
1998	97.00	32.33
1998	97.00	64.67
1999	97.00	97.00
1999	98.00	97.33
1999	95.00	96.67
2000	95.00	96.00
2000	96.00	95.33

In the versions up to 4.1.4 the Previous function does not see any value before Row 1 so it places the value 97 in the cell and the same for Row 2.

In version 4.15 an later the Previous function looks for a value in the table. If there is no previous value to Row 1, it sees this as a NULL value. The calculation then becomes:
Row 1: (97+0+0) / 3 = 32.
Row 2: (97+ 97+0) / 3 = 64.

If this is not required you have to make sure that the first two rows are taken out of the Previous calculation and force BusinessObjects to take into account the NULL values.

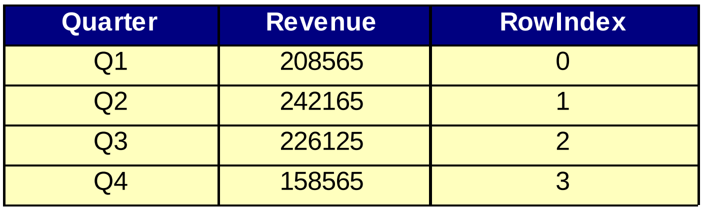
To do so, use the formula below:

= If IsNull(Previous((<Value> in <Year>))) Or IsNull(Previous(Previous((<Value> in <Year>)))) Then <Value> Else ((<Value> In <Year>)+Previous((<Value> In <Year>))+Previous(Previous((<Value> In <Year>)))) /3

Note: If there is no previous value, insert the original measure value. Otherwise calculate the rolling average

8.8 RowIndex

The RowIndex( ) Function returns the row numbers, starting from 0, of all rows in the report.
Note that the row index does not include the header or the footer of the report.



Quarter	Revenue	RowIndex
Q1	208565	0
Q2	242165	1
Q3	226125	2
Q4	158565	3

This Function provides a unique row number for each row even if rows are aggregated in the report. For example with importing an Excel file with numeric values. The numeric value automatically gets set to a measure with a sum aggregation, the data gets ‘added’ up to produce one row. By inserting a new column with the RowIndex Function this will force the data to return all the rows. (If the Rowindex is set to a variable, you can later hide this in Format /Table/ Pivot)

This is also the case with the crosstabs, where you can not use the option ‘avoid duplicate row aggregation’ the RowIndex Function will bring back all rows.

Another tip is that when you are calculating a Count on a dimension in Business Objects, this calculates a Count Distinct. In order for it to calculate all values, even the duplicate, create a variable with the Function Rowindex, insert this in the report, and formulate the Count Function as follows:

=Count(<object>) In <RowIndex Variable>

9 Additional Functions in V5.x

Aggregates

Count All
Product
RunningProduct

Numeric Functions

EuroConvertFrom
EuroConvertTo
EuroFromRoundErr
EuroToRoundErr
Median

Logical Functions

Even
Odd

Document Functions

DrillFilters
GlobalFilters
PageInSection

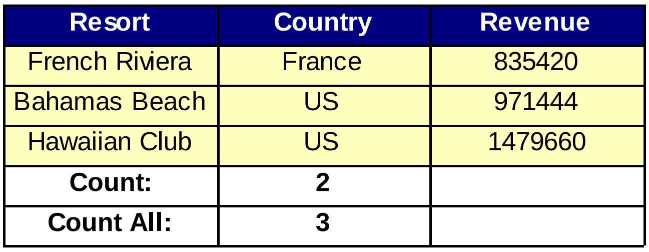
Misc Functions

MultiCube

9.1 Aggregates

9.1.1 Count All

The CountAll Function counts the number of rows in the body, including all duplicates and empty rows.



Resort	Country	Revenue
French Riviera	France	835420
Bahamas Beach	US	971444
Hawaiian Club	US	1479660
Count:	**2**	
Count All:	**3**	

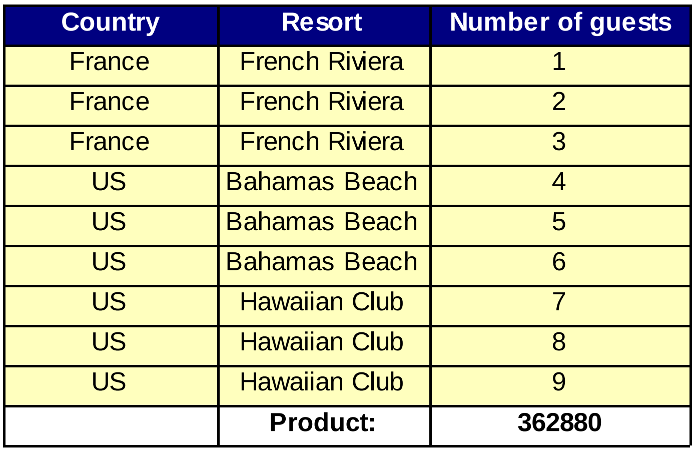
In v4 there is only the option to use count, which will count only unique values when applied to a dimension, and will not count each individual value.
As US is a duplicate this is shown as count 2 rather than count 3, whereas the CountAll will bring back the value 3.

9.1.2 Product

The "Product" function is an Aggregate Function that returns the total multiplication of all the numbers given in the column where the function is applied on.

NOTE: This Function is only available in V5.1 and after

For Example, when applying the function at the sum field of a table with the values 1,2&3 it will give the value of 6 (1*2*3=6)
The Product Function can be applied on dimensional numeric values, or measures.



Country	Resort	Number of guests
France	French Riviera	1
France	French Riviera	2
France	French Riviera	3
US	Bahamas Beach	4
US	Bahamas Beach	5
US	Bahamas Beach	6
US	Hawaiian Club	7
US	Hawaiian Club	8
US	Hawaiian Club	9
	Product:	**362880**

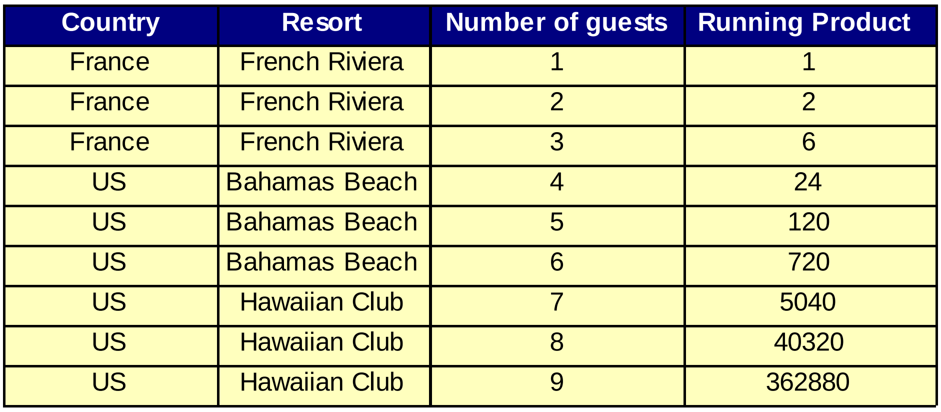
The extended syntax for Aggregate Functions allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.
An example for this with the Product Function is the calculation of the Product of Number of Guests per Country:
=Product(<Number of guests>) In <Country>

9.1.3 RunningProduct

The RunningProduct is an Aggregate Function that returns the running total multiplication of all the numbers given in the column where the function is applied on.

NOTE: This function is only available in V5.1 and after

For Example, when applying the function at the sum field of a table with the values 1,2&3 it will give the value of 1 (1*0),
2 (1*2)& 6 (2*3)
The RunningProduct function can be applied on dimensional numeric values, or measures.



Country	Resort	Number of guests	Running Product
France	French Riviera	1	1
France	French Riviera	2	2
France	French Riviera	3	6
US	Bahamas Beach	4	24
US	Bahamas Beach	5	120
US	Bahamas Beach	6	720
US	Hawaiian Club	7	5040
US	Hawaiian Club	8	40320
US	Hawaiian Club	9	362880

The first value will be repeated in the RunningProduct column as it will be a calculation of: (1*empty)=1. The second value will be (1*2)= 2.The third value will be (1*2*3)=6. The fourth value: (1*2*3*4) =24.

The extended syntax for this Aggregate Function allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators.

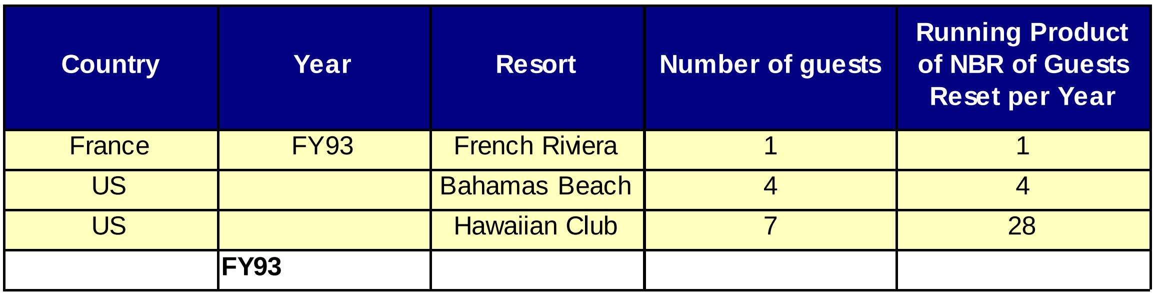
An Example for this is:

=RunningProduct(<Number Of Guests>) In <Country>

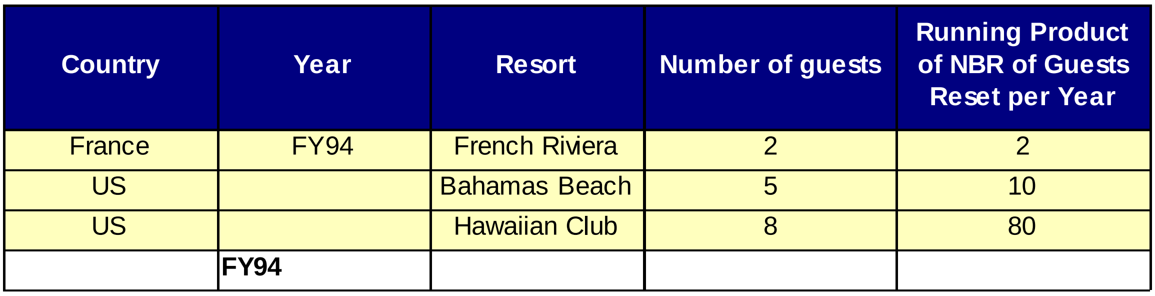
Country	Year	Resort	Number of guests	Running Product of NBR of Guests per Country
France	FY93	French Riviera	1	6
France	FY94	French Riviera	2	6
France	FY95	French Riviera	3	6
US	FY93	Bahamas Beach	4	362880
US	FY94	Bahamas Beach	5	362880
US	FY95	Bahamas Beach	6	362880
US	FY93	Hawaiian Club	7	362880
US	FY94	Hawaiian Club	8	362880
US	FY95	Hawaiian Club	9	362880

As any other RunningAggregate you can make it reset per break or section or per row or column.
=RunningProduct(<Number Of Guests>;<Year>) when resetting per break of the object <Year >

Or =RunningProduct(<Number Of Guests>;Col<Year>) when resetting per column in a Crosstab.



Country	Year	Resort	Number of guests	Running Product of NBR of Guests Reset per Year
France	FY93	French Riviera	1	1
US		Bahamas Beach	4	4
US		Hawaiian Club	7	28
	FY93			



Country	Year	Resort	Number of guests	Running Product of NBR of Guests Reset per Year
France	FY94	French Riviera	2	2
US		Bahamas Beach	5	10
US		Hawaiian Club	8	80
	FY94			

If the formula needs to be extended further, all that needs to be added is a comma, followed by a new dimensional object.
=RunningAggregate(<Measure>;<Dimension1>,<Dimension2>)

9.2 Numeric Functions

9.2.1 EuroConvertFrom

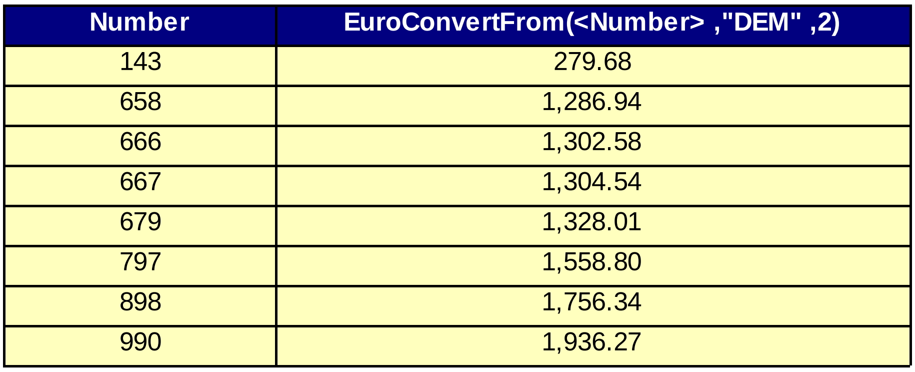
The EuroConvertFrom Function converts an amount in euros into a specified currency (Formula:EuroConvertFrom(value, currency ISO code, number of decimals)).

The currency ISO code is the ISO code for the target currency.
In the number of decimals argument, you indicate how many decimal digits you want to display in the result.
Options are 0-15.
Example 1:

=EuroConvertFrom(1,234.56, “DEM”, 2)
This converts 1,234.56 euros to German marks. The result is rounded and displayed with 2 decimal digits. The result is 1,234.56 x 1.95583 = 2,414.59.

Example 2:

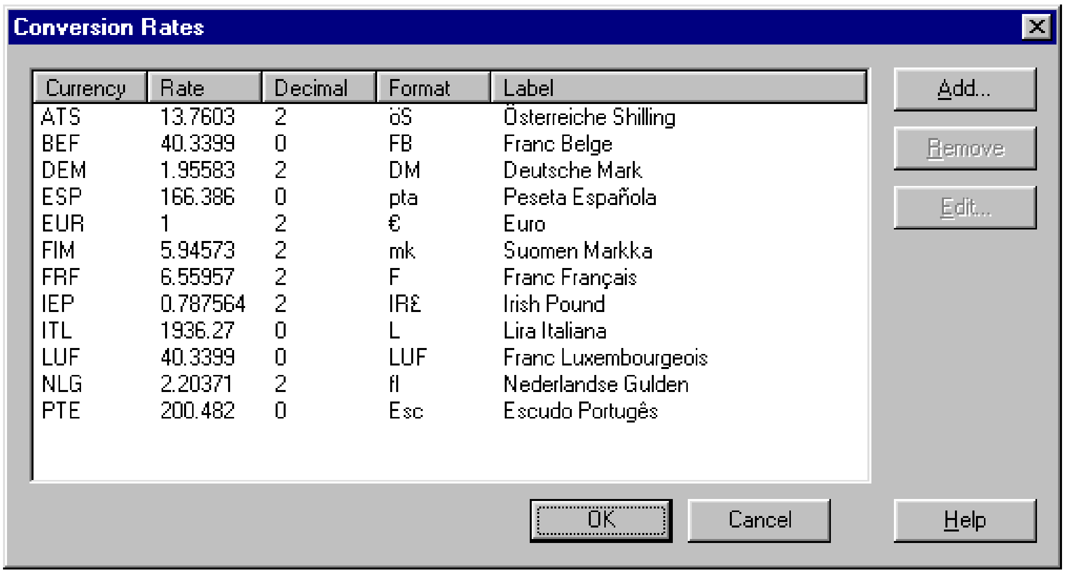
=EuroConvertFrom(<Sales Revenue>, “FRF”, 2)
This converts all values of the measure <Sales Revenue> from euros to French francs and displays the result with two decimal digits.



Number	EuroConvertFrom(<Number> ,"DEM" ,2)
143	279.68
658	1,286.94
666	1,302.58
667	1,304.54
679	1,328.01
797	1,558.80
898	1,756.34
990	1,936.27

In order to check up on the conversion values, there is a special menu in V5 to show this.
In the Data menu in the Taskbar, there is a menu called Euro.

When selecting the option “Display Convertion rates”, Business Objects will present you with the following menu:



Currency	Rate	Decimal	Format	Label
ATS	13.7603	2	öS	Österreiche Shilling
BEF	40.3399	0	FB	Franc Belge
DEM	1.95583	2	DM	Deutsche Mark
ESP	166.386	0	pta	Peseta Española
EUR	1	2	€	Euro
FIM	5.94573	2	mk	Suomen Markka
FRF	6.55957	2	F	Franc Français
IEP	0.787564	2	IR£	Irish Pound
ITL	1936.27	0	L	Lira Italiana
LUF	40.3399	0	LUF	Franc Luxembourgeois
NLG	2.20371	2	fl	Nederlandse Gulden
PTE	200.482	0	Esc	Escudo Portugês

You can hereby check if the formula values given are accurate. The Euro Menu will also allow you to use the option Convert From Euros (when highlighting a report column) , which does the same as the function EuroConvertFrom , but will not allow you to set the number of Decimal Digits that are required. These have to be adjusted in Format Cell.

With the EuroConvertFrom function, this can also be formatted with the currency before the value via:

=FormatNumber(EuroConvertFrom(<value> ,"DEM" ,2) ,"DM 0,0")

This will give for example: “DM 1,070,766”

9.2.2 EuroConvertTo

The EuroConvertTo Function converts an amount from a given currency into euros. (Formula: EuroConvertTo(value, currency ISO code, number of decimals)

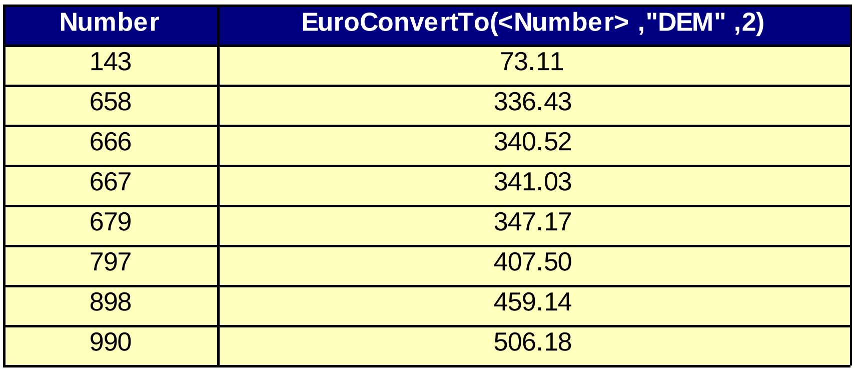
The currency ISO code is the official ISO code for the currency you are converting from. In the number of decimals argument, you indicate how many decimal digits you want to display in the result. Options are 0-15.

Example 1:

=EuroConvertTo(1,234.56, DEM, 2) converts 1,234.56 German marks to euros. The result is rounded and displayed with 2 decimal digits. The result is 1,234.56 / 1.95583 = 624.73

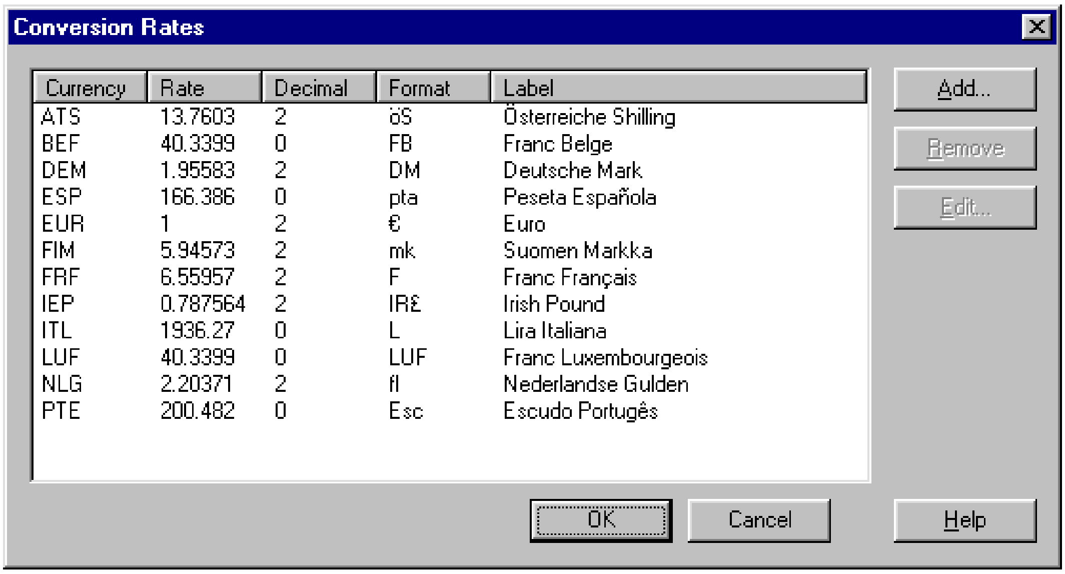
Example 2:

=EuroConvertTo(<Sales Revenue>, FRF, 2) converts all values of the measure <Sales Revenue> from French francs into euros and displays the result with two decimal digits.



Number	EuroConvertTo(<Number> ,"DEM" ,2)
143	73.11
658	336.43
666	340.52
667	341.03
679	347.17
797	407.50
898	459.14
990	506.18

In order to check up on the conversion values, there is a special menu in V5 to show this.
In the Data menu in the Taskbar, there is a menu called Euro. When selecting the option “Display Convertion rates”, Business Objects will present you with the following menu:



Currency	Rate	Decimal	Format	Label
ATS	13.7603	2	öS	Österreiche Shilling
BEF	40.3399	0	FB	Franc Belge
DEM	1.95583	2	DM	Deutsche Mark
ESP	166.386	0	pta	Peseta Española
EUR	1	2	€	Euro
FIM	5.94573	2	mk	Suomen Markka
FRF	6.55957	2	F	Franc Français
IEP	0.787564	2	IR£	Irish Pound
ITL	1936.27	0	L	Lira Italiana
LUF	40.3399	0	LUF	Franc Luxembourgeois
NLG	2.20371	2	fl	Nederlandse Gulden
PTE	200.482	0	Esc	Escudo Portugês

You can hereby check if the formula values given are accurate. The Euro Menu will also allow you to use the option Convert To Euros (when highlighting a report column), which does the same as the function EuroConvertTo, but will not allow you to set the number of Decimal Digits that are required. These have to be adjusted in Format Cell. (This will also allow you to add the Euro sign)

9.2.3 EuroFromRoundErr

The EuroFromRoundErr function returns the difference between the number converted from euros to a target currency and rounded to the specified number of decimals and the same number before rounding. Returns a number in the specified currency. The currency ISO code is the character string that indicates the currency ISO code for the source currency. The third argument indicates the number of decimal digits to display. It ranges from 0 to 15.

For Example:

=EuroFromRoundErr(1,234.56, DEM, 2) returns the rounding error when you convert 1,234.56 euros to German marks and display the result with two decimal digits.

Number	EuroFromRoundErr(<Number> ,"DEM" ,2)
143	-0.00
658	0.00
666	-0.00
667	0.00
679	0.00
797	0.00
898	0.00
990	-0.00

9.2.4 EuroToRoundErr

The EuroToRoundErr function returns the difference between the number converted to euros and rounded to the specified number of decimals and the same number before rounding. Returns a number in euros. The currency ISO code is the character string that indicates the currency ISO code for the source currency. The third argument indicates the number of decimal places to display. It ranges from 0 to 15.

For Example:

=EuroToRoundErr(1,234.56, DEM, 2) returns the rounding error when you convert 1,234.56 German marks to euros and display the result with two decimal digits.

Number	EuroToRoundErr(<Number> ,"DEM" ,2)
143	-0.00
658	-0.00
666	-0.00
667	-0.00
679	0.00
797	0.00
898	-0.00
990	0.00

9.2.5 Median

The Median Function returns the median of a given set of numeric values. The median is the middle number in the set. Logical and empty values are ignored, zero is included.

Example 1:

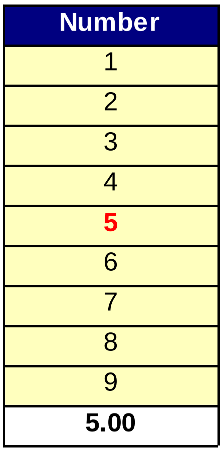
If an Object contains the values {1, 2, 3, 4, 5}
The Formula: =Median(<Object>) returns 3.

Example 2:

If there is an even number of numbers in the set, the Median function calculates the average of the two numbers in the middle.
If the Object contains the values {1, 2, 3, 4, 5,6} then the Formula: =Median(<Object>) returns 3.5
The average of 3 and 4

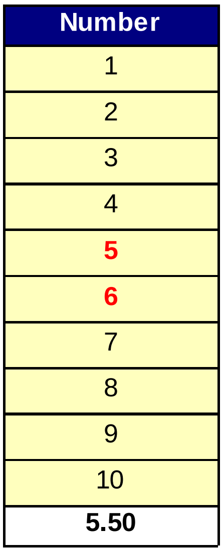
Note that this function is used for numeric objects, which are measures (not dimensions (this will give #computation errors), with aggregation set to “none”

Odd:



Number
1
2
3
4
5
6
7
8
9
5.00

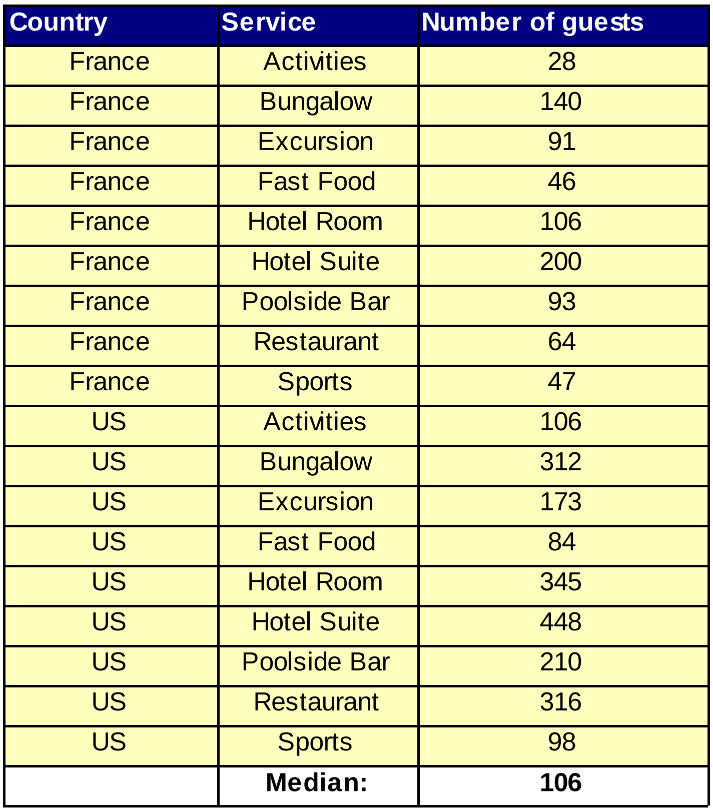
Even:



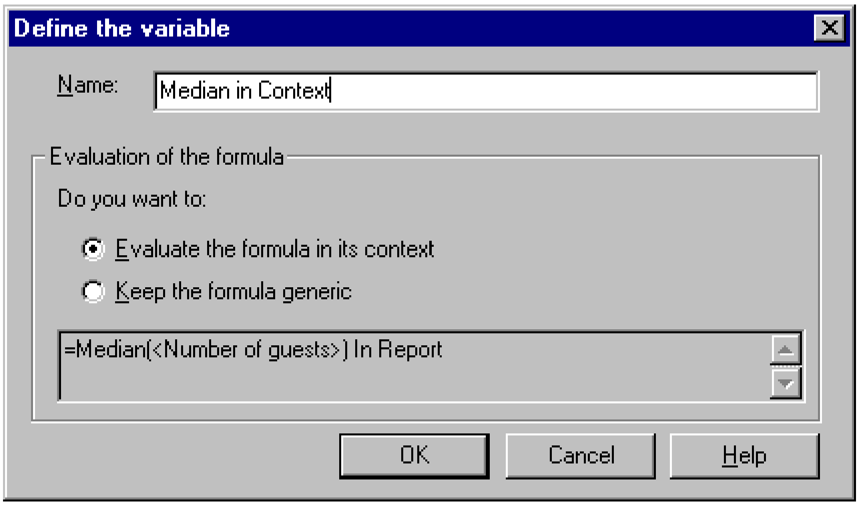
Number
1
2
3
4
5
6
7
8
9
10
5.50

Note: The extended syntax for this aggregate function allows you to further delimit the context on which the function is to be applied. The context can be specified with In, ForEach, or ForAll operators. In the case of the Median Function there is however a special technique to be followed.

If one wishes to indicate the Median result in a report column, one needs to first highlight the white field in which the Median is calculated. Go then to the Data Menu in the Taskbar and select the option Define as Variable. Give it a new name and tick the option “evaluate formula in context”



Country	Service	Number of guests
France	Activities	28
France	Bungalow	140
France	Excursion	91
France	Fast Food	46
France	Hotel Room	106
France	Hotel Suite	200
France	Poolside Bar	93
France	Restaurant	64
France	Sports	47
US	Activities	106
US	Bungalow	312
US	Excursion	173
US	Fast Food	84
US	Hotel Room	345
US	Hotel Suite	448
US	Poolside Bar	210
US	Restaurant	316
US	Sports	98
	Median:	**106**



Define the variable

Name: Median in Context

Evaluation of the formula

Do you want to:

● Evaluate the formula in its context

○ Keep the formula generic

=Median(<Number of guests>) In Report

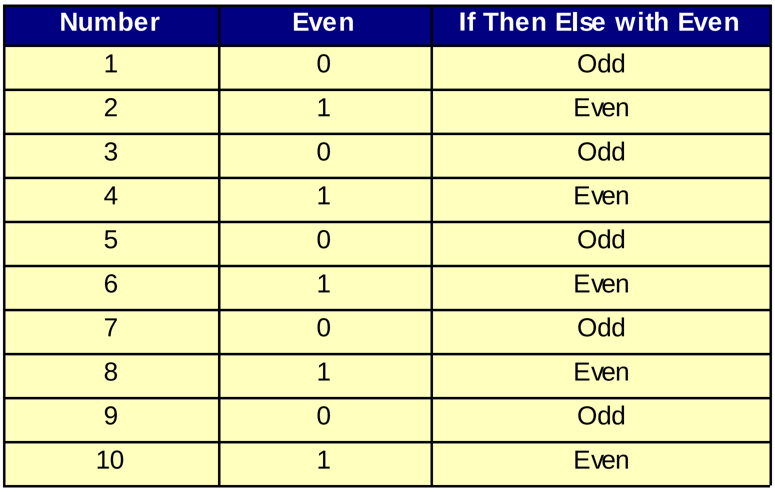
This will now allow you to insert this in an additional column in the report.

9.3 Logical Functions

9.3.1 Even

The Even Function returns a 1 for an even number and a 0 for an odd number.

This can be used in combination with IF Then Else.
For example:
= If Even(<number>) Then "Even" Else "Odd"

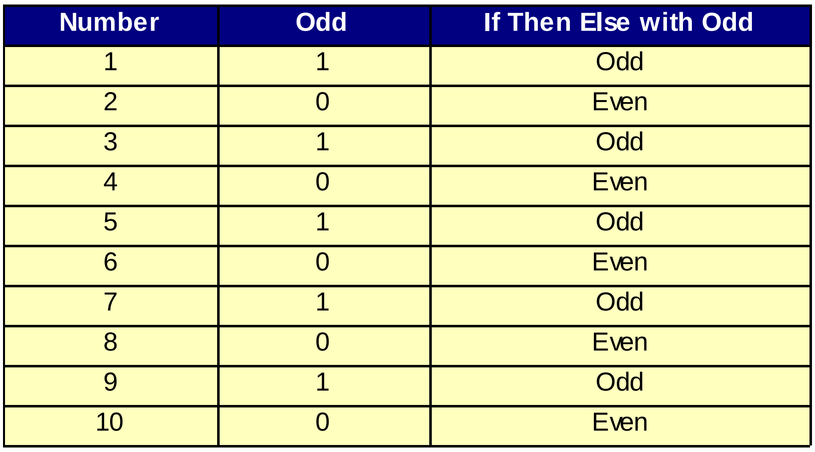


Number	Even	If Then Else with Even
1	0	Odd
2	1	Even
3	0	Odd
4	1	Even
5	0	Odd
6	1	Even
7	0	Odd
8	1	Even
9	0	Odd
10	1	Even

9.3.2 Odd

The Odd Function returns a 1 for an odd number and a 0 for an even number.

This can be used in combination with IF Then Else.
For example:
= If Odd(<number>) Then "Odd" Else "Even"



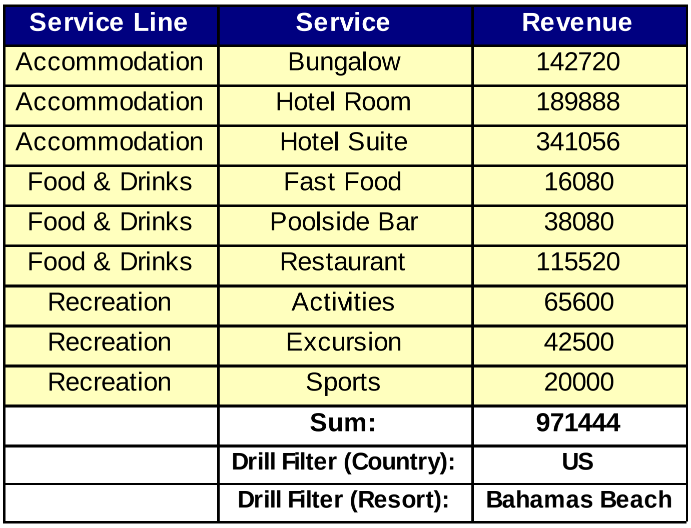
Number	Odd	If Then Else with Odd
1	1	Odd
2	0	Even
3	1	Odd
4	0	Even
5	1	Odd
6	0	Even
7	1	Odd
8	0	Even
9	1	Odd
10	0	Even

9.4 Document Functions

9.4.1 DrillFilters

The DrillFilters Function returns the filtered values of the variable(s) filtered in drill mode.

For Example:



Service Line	Service	Revenue
Accommodation	Bungalow	142720
Accommodation	Hotel Room	189888
Accommodation	Hotel Suite	341056
Food & Drinks	Fast Food	16080
Food & Drinks	Poolside Bar	38080
Food & Drinks	Restaurant	115520
Recreation	Activities	65600
Recreation	Excursion	42500
Recreation	Sports	20000
	Sum:	**971444**
	Drill Filter (Country):	**US**
	Drill Filter (Resort):	**Bahamas Beach**

In drill mode, you have drilled down on Country and Resort. Your block currently displays data for US and Bahamas Beach. If you enter the formula =DrillFilters(<Country>) in a cell, it returns US.

In order to display the other values that have been selected to drill on you need to extend the formula:

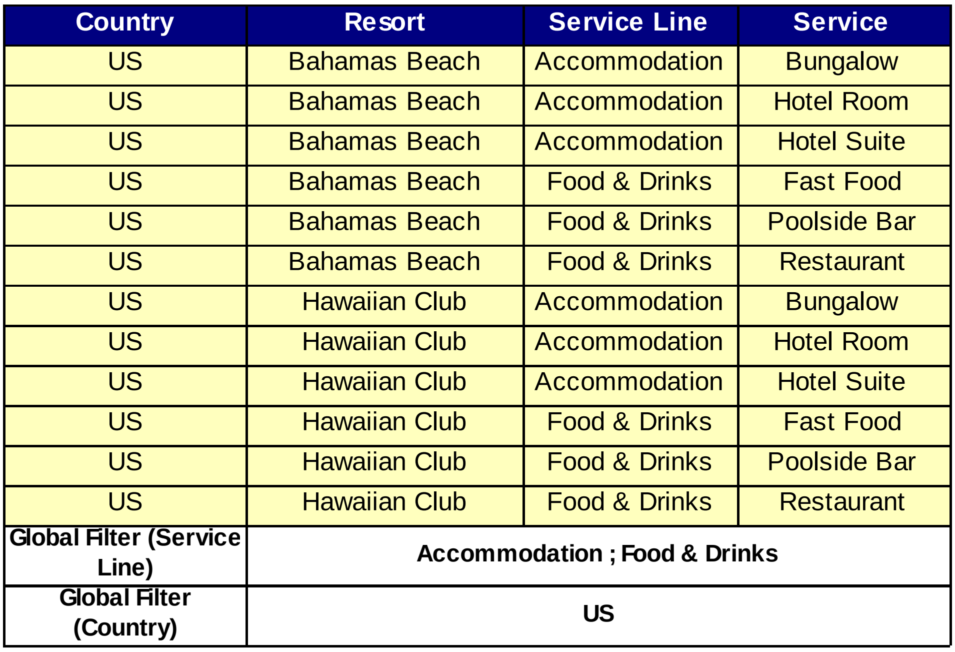
=DrillFilters(<Country>)&" "&DrillFilters(<Resort>)&" "&DrillFilters(<Service>)

This will return: “US Bahamas Beach Accommodation”

9.4.2 GlobalFilters

The GlobalFilters Function returns the filtered values of the specified variable(s).
Note: only filters that are applied Globally (based on the whole report) and not Table filters. That means that filters that have been applied via the filter icon will not show up, as these are not global filters.
Global filers can be applied in the Format Filter menu.

For Example:



Country	Resort	Service Line	Service
US	Bahamas Beach	Accommodation	Bungalow
US	Bahamas Beach	Accommodation	Hotel Room
US	Bahamas Beach	Accommodation	Hotel Suite
US	Bahamas Beach	Food & Drinks	Fast Food
US	Bahamas Beach	Food & Drinks	Poolside Bar
US	Bahamas Beach	Food & Drinks	Restaurant
US	Hawaiian Club	Accommodation	Bungalow
US	Hawaiian Club	Accommodation	Hotel Room
US	Hawaiian Club	Accommodation	Hotel Suite
US	Hawaiian Club	Food & Drinks	Fast Food
US	Hawaiian Club	Food & Drinks	Poolside Bar
US	Hawaiian Club	Food & Drinks	Restaurant
Global Filter (Service Line)	**Accommodation ; Food & Drinks**		
Global Filter (Country)	**US**		

You have filtered a report to display data only for the US. To do this, you placed the Country dimension in the Global Filters folder in the filters dialog box and selected the value US.
f you enter the formula GlobalFilters(<Country>) in a cell, it returns "US".

Or when there are more values used in the filter, these will be displayed in the cell, separated by a semi-colon.

9.4.3 PageInSection

The PageInSection Function resets the page number to 1 at the start of each new section in the specified section level.

For Example:

You have a report with two section levels, Country and Quarter. You set every value of Country to start on a new page and enter the following syntax in a cell placed WITHIN THE SECTION THAT NEEDS THE RESETTIG OF THE PAGE (it can not be placed in the header or footer, as it can then not refer to the relevant section. It will give the value 0 instead), to restart page numbering to 1 for each value of country:

=PageInSection(1)

Where (1) is Country which is section 1 in the report.
If there are any following sections these can also be reset via this formula with the use of the relevant section number.

This option can be used effectively when you set each value of a section to start on a new page. To do this, check the Start on a new page option in the Section Format dialog box.

Do remember that the result will only be seen in the report, when the option View / Page Layout has been set, otherwise the result will be seen as a 0 at each section (this however does get correctly displayed in the Print Preview, and when printing without the page layout option set)

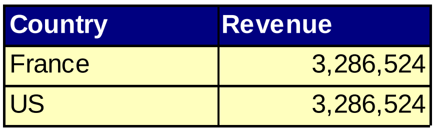
9.5 Misc Functions

9.5.1 MultiCube

With the MultiCube function calculation can be made possible when using objects from several linked dataproviders in one report.

For Example:

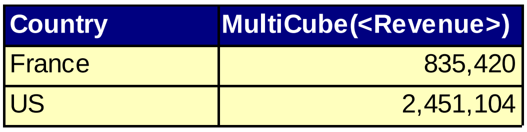
You have two Queries : Query 1 with Country, and Resort. Query 2 with Resort and Revenue. The two queries are linked via the object Resort. When now running a query with Country (q1) , and Revenue (q2) this will show the wrong totals.



Country	Revenue
France	3,286,524
US	3,286,524

The reason for this is that the these two objects are not compatible as BO does not know which part of the revenue belongs to which country There is only a link between resort and BO has not run the revenue values against the country values.
We can make the logical assumption here that “Bahamas Beach” and “Hawaiian Club” are resorts belonging to the “US” and that “French Riviera” belongs to “France”, but BO can not make this assumption.
The function MultiCube helps to force this combination of objects to work.

The function: =MultiCube(<Revenue>) would make sure that the Revenue values of the resorts are aggregated according to the country they belong to.



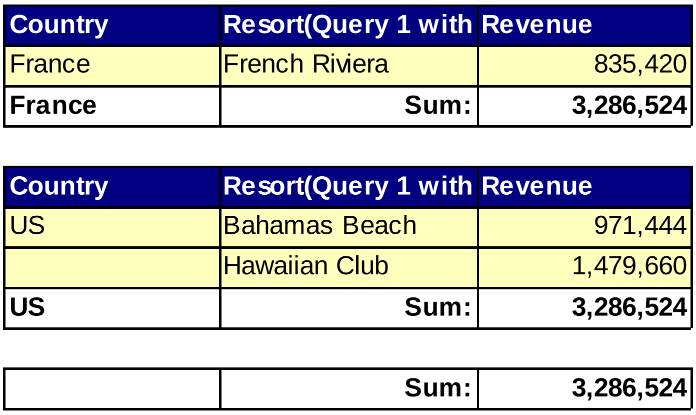
Country	MultiCube(<Revenue>)
France	835,420
US	2,451,104

The function has its limitations with regards to logic
For example: Q1 has Year, Months and Revenue. Q2 Year and Quarter.
The queries are linked via the object Year.
When you now combine Quarter & Revenue it will give the wrong results. For us it is logic to see that January, February and March are part of Q1, BO can not know this.
The function MultiCube helps to force this combination of objects to work and can make a aggregation of the revenue values of January, February and March for Q1.It will give the right results back
However, when this is run the other way it will not work:
Query 1 with Year, Quarter and Revenue. Query 2 with Year and Month. If you now run a query with Month & Revenue there is no way that you can tell which part of the Revenue for Q1 is to belong to Feb, Jan or March. The Multicube function can therefore not solve this equation either.

The same function will allow to get the correct Break and Section totals in a synchronised report, with a non-linked object set as a break or master.

For Example :

You have two queries. Query 1 returns <Country > and <Resort>, and Query 2 returns <Resort > and <Revenue> . When linking on <Resort> and then inserting a new query with <Country > ,<Revenue> , <Resort (Query 1 with Beach)> and apply a break on <Country> this will display the correct values in the column, but when trying to sum the values at break level Business Objects produced the Grand Total again.



Country	Resort(Query 1 with	Revenue
France	French Riviera	835,420
France	**Sum:**	**3,286,524**

Country	Resort(Query 1 with	Revenue
US	Bahamas Beach	971,444
	Hawaiian Club	1,479,660
US	**Sum:**	**3,286,524**

	Sum:	**3,286,524**

The reason of the problem is the same as the previous example; the sum calculation will give the wrong total as Business Objects can not see the which part of the revenue goes with which country, and will give the grand total at each break level.

To resolve this you need to highlight the sum field at the break level and replace the Sum(<Revenue>)field with the formula :

=MultiCube(<Revenue>)
in which case it no longer needs a sum ,
or you can place the MultiCube on the Revenue Column, and then apply a sum to this :

=Sum((MultiCube(<Revenue(Query 1 with BEACH)>)))

Also note that in some reports with several data providers you may have to first apply the formula on the report column , replacing for example the <Revenue > column by
:=MultiCube(<Revenue>)
and then highlight the sum field and apply the same formula there.
In cases where this is found necessary, the formula
=MultiCube(<Revenue>)
At the sum level would insert only the first value from the <Revenue> per break.