BusinessObjects Context Formulas: Making Sense of ForEach, ForAll, and In

Presented by
Brie Anne Clark

CREATIVE TECHNOLOGY
& TRAINING SOLUTIONS

http://www.cttsbi.com

Context Basics

There are three operators in BusinessObjects that can be used to alter the context in which a formula is evaluated - **ForEach**, **ForAll**, and **In**. When you create a formula without specifying a context, BusinessObjects determines the context based on the formula's location in the report. To demonstrate this, create a report based on the Island Resorts Marketing universe with the **Resort**, **Service Line**, and **Revenue** objects. Then, apply a break on **Resort** and add a sum and an average for **Revenue**. In this case, the sum of **Revenue** is calculated for each **Resort** and the average of **Revenue** is based on the totals for the individual **Service Lines**.

Resort	Service Line	Revenue
Bahamas Beach	Accommodation	$673,664
	Food & Drinks	$169,680
	Recreation	$128,100
Bahamas Beach Total		**$971,444**
Bahamas Beach Average		**$323,815**

Remove **Service Line** from the block. When you do this, you are removing it from the calculation context. This means that the sum stays the same, but the average is now based on each resort. Since there is a break on **Resort**, the average is the same as the sum.

Resort	Revenue
Bahamas Beach	$971,444
Bahamas Beach Total	**$971,444**
Bahamas Beach Average	**$971,444**

If you want to see the **Service Line** average, you must add it to the context of the formula. To do this, use the following syntax:

=Average(<Revenue> ForEach <Service Line>)

By using this formula syntax, the calculation produces the following output:

Resort	Revenue
Bahamas Beach	$971,444
Bahamas Beach Total	**$971,444**
Bahamas Beach Average	**$323,815**

The ForEach operator allows you to add a dimension to the calculation context, as demonstrated in the example above. There are many uses for this operator, just as there are many uses for the ForAll operator. The ForAll operator allows you to remove dimensions from the calculation context.

Building on the first example, add the Service dimension to the original table block to see the following output:

Resort	Service Line	Service	Revenue
Bahamas Beach	Accommodation	Bungalow	$142,720
	Accommodation	Hotel Room	$189,888
	Accommodation	Hotel Suite	$341,056
	Food & Drinks	Fast Food	$16,080
	Food & Drinks	Poolside Bar	$38,080
	Food & Drinks	Restaurant	$115,520
	Recreation	Activities	$65,600
	Recreation	Excursion	$42,500
	Recreation	Sports	$20,000
	Bahamas Beach Total		**$971,444**
	Bahamas Beach Average		**$107,938**

Notice that the average **Revenue** cell is now calculating the average based on the individual services at the **Resort**. What if you still want an average that is based on the **Service Line** totals, but you don't want to physically remove **Service** from the table block? Use the following syntax to remove **Service** from the calculation context:

=Average(<Revenue> ForAll <Service>)

By using this formula syntax, the calculation produces the following output:

Resort	Service Line	Service	Revenue
Bahamas Beach	Accommodation	Bungalow	$142,720
	Accommodation	Hotel Room	$189,888
	Accommodation	Hotel Suite	$341,056
	Food & Drinks	Fast Food	$16,080
	Food & Drinks	Poolside Bar	$38,080
	Food & Drinks	Restaurant	$115,520
	Recreation	Activities	$65,600
	Recreation	Excursion	$42,500
	Recreation	Sports	$20,000
	Bahamas Beach Total		**$971,444**
	Bahamas Beach Average		**$323,815**

The third operator that can be used for modifying a formula's calculation context is In. For the first two examples, instead of using the ForEach or ForAll operators, you could have used the following syntax:

=Average(<Revenue> In (<Service Line> , <Resort>))

Notice that the same syntax works in either example because rather than adding dimensions to or removing dimensions from the current context, you are specifying the precise context. Thus, your results should not change when additional objects are added to the table block, as shown below:

Resort	Year	Service Line	Service	Revenue
Bahamas Beach	FY01	Accommodation	Bungalow	$37,600
	FY01	Accommodation	Hotel Room	$64,320
	FY01	Accommodation	Hotel Suite	$111,544
	FY01	Food & Drinks	Fast Food	$4,640
	FY03	Food & Drinks	Fast Food	$5,640
	FY03	Food & Drinks	Poolside Bar	$14,280
	FY03	Food & Drinks	Restaurant	$75,335
	FY03	Recreation	Activities	$22,400
	FY03	Recreation	Excursion	$15,300
	FY03	Recreation	Sports	$8,000
		Bahamas Beach Total		**$971,444**
		Bahamas Beach Average		**$323,815**

In the above example, the In operator was used to specify which dimensions to use in the calculation context. It can also be used with the words Block, Body, and Report. For example, if a block contains Resort, Service Line, and Revenue and there is a break on Resort, using the In operator with the word Body in the break footer refers to the rows for a particular resort. If a block contains Resort, Service Line, and Revenue and the report is sectioned by Country, using the In operator with the word Block refers to the rows for a particular country. Finally, using the In operator with the word Report is referring to all of the rows in the report.

Shading Above Average Rows

Problem:

I only want to shade rows that are above the section average.

Solution:

1. Create a new report using **Year**, **Month Name**, and **Sales revenue** from the **eFashion** universe.

2. Section the report by **Year** and apply a custom sort on **Month Name** so that the column is sorted in chronological order.

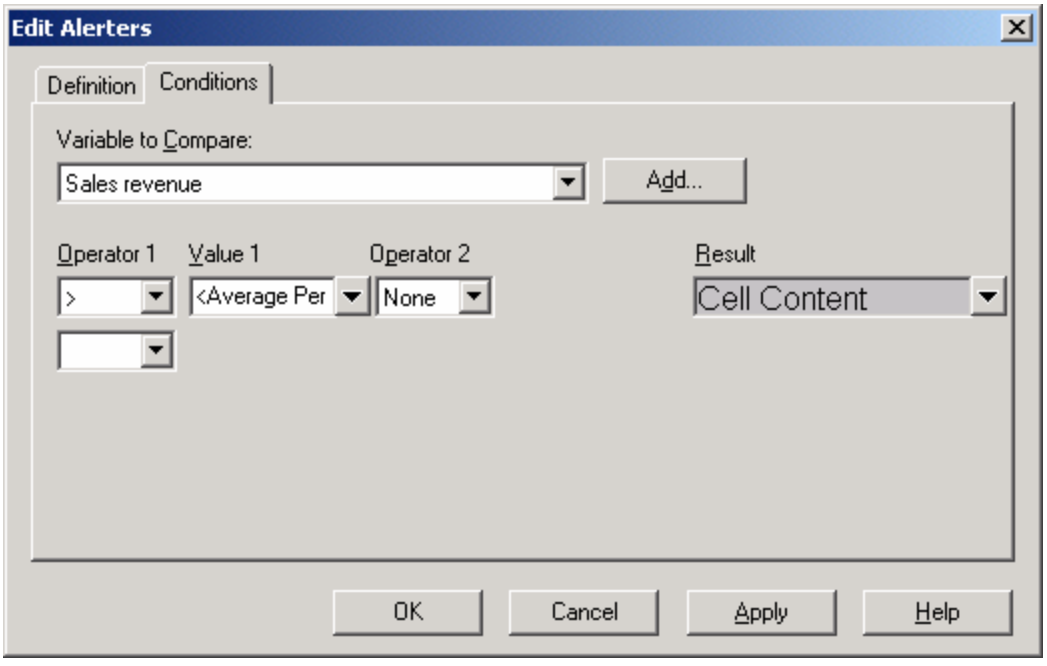
3. Create a variable named **Average Per Year** with the following formula:

   ```
   =Average(<Sales revenue>) In (<Year>)
   ```

4. Create a new alerter named **Above Average Shading**.

5. On the **Conditions** tab, select **Sales revenue** from the Variable to Compare drop-down list.

6. From the **Operator 1** drop-down list, select **> (greater than)**. From the **Value 1** drop-down list, select **Variables**, select **Average Per Year**, and then click **Insert**. From the **Operator 2** drop-down list, select **None**. Format the result to have **black** text and a **gray** fill color.



7. Click **OK** to return to the Alerters dialog box.

8. Apply the alerter to the body of the report. Your report should look similar to the one below.

2002

Month Name	Sales revenue
January	$1,003,541
February	$630,073
March	$1,027,085
April	$895,260
May	$865,615
June	$517,819
July	$525,904
August	$173,756
September	$668,181
October	$655,206
November	$484,024
December	$649,350

Shading a Maximum Row

Problem:

I only want to shade rows that are above the section average.

Solution:

1. Create a new report using **Year**, **Month**, **Lines**, and **Sales revenue** from the **eFashion** universe.

2. Section the report by **Year** and apply a break on **Lines**.

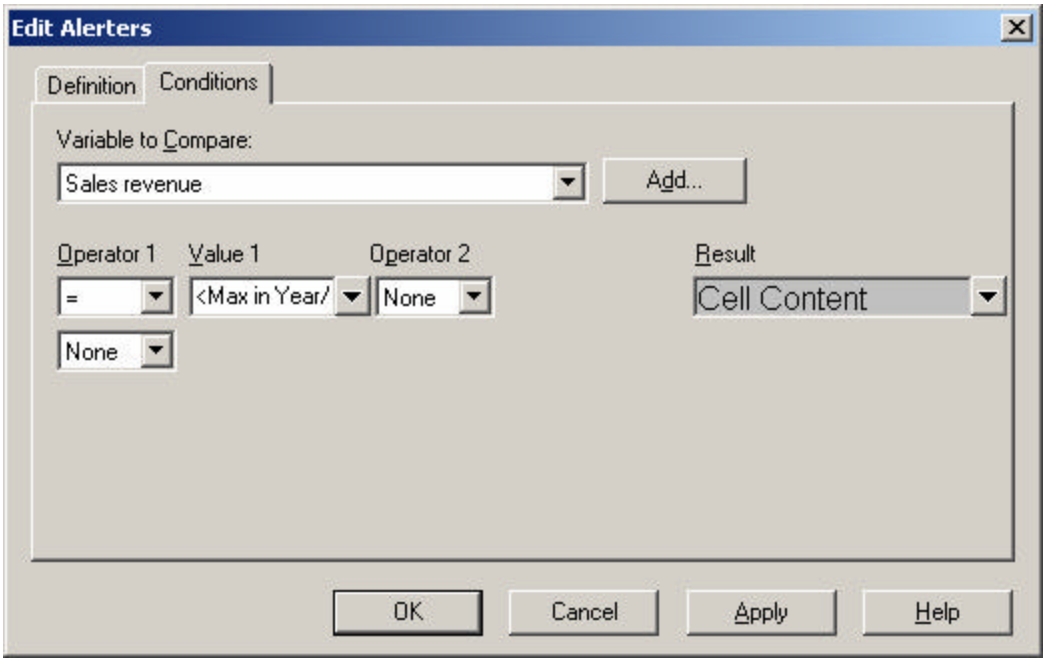
3. Create a variable named **Max in Year/Lines** with the following formula:

   ```
   =Max(<Sales revenue>) In (<Year>,<Lines>)
   ```

4. Create a new alerter named **Shade Max in Year/Lines**.

5. On the **Conditions** tab, select **Sales revenue** from the Variable to Compare drop-down list.

6. From the **Operator 1** drop-down list, select **= (equal to)**. From the **Value 1** drop-down list, select **Variables**, select **Max in Year/Lines**, and then click **Insert**. Format the result to have **black** text and a **gray** fill color.



7. Click **OK** to return to the Alerters dialog box.

8. Apply the alerter to the body of the report. Your report should look similar to the one below.

2002

Lines	Month	Sales revenue
Accessories	1	$379,595
	2	$252,697
	3	$505,836
	4	$431,772
	5	$382,521
	6	$220,780
	7	$218,885
	8	$48,159
	9	$40,060
	10	$29,160
	11	$10,215
	12	$26,543

Highlighting Table Blocks That Are Over a Section Average

Problem:

I want above average blocks in my report to stand out, but I do not want to calculate the average in the context of the dimensions in the table block.

Solution:

1. Create a new report using **Year**, **Quarter**, **State**, and **Quantity sold** from the **eFashion** universe.

2. Section the report by **Year** and apply a break on **Quarter**.

3. Enter the following formula in the break footer to display the average of the quarters within the year:

   ```
   =Average(<Quantity sold> ForAll <State>) In Block
   ```

4. Next, you will need a variable that determines whether the **Quarter** total is above or below the average for the **Year**. Create a new variable named **Over or Under Average** with the following formula:

   ```
   =If(Sum(<Quantity sold>) - Average(<Quantity sold> ForAll <State>)
   In Block) > 0 Then "Over" Else "Under"
   ```

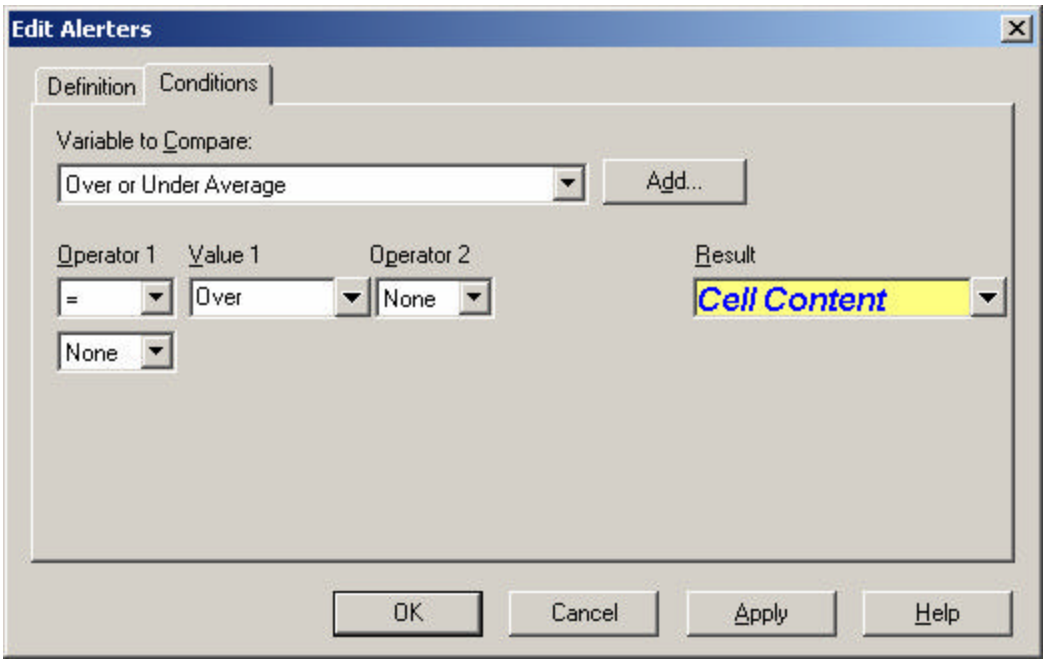
5. Next, create a label in the break footer that concatenates the result of the **Over or Under Average** variable with some descriptive text:

   ```
   =<Quarter> + " Was " + <Over or Under Average> + " Average for " +
   <Year>
   ```

6. Now that the average has been calculated, the variable that displays "Over" or "Under" has been created, and the label has been entered, you can apply an alerter to the label to make it stand out when the total for the **Quarter** is above the average for the **Year**. To do this, create a new alerter named **Highlight Over Average**.

7. On the **Conditions** tab, select **Over or Under Average** from the Variable to Compare drop-down list.

8. From the **Operator 1** drop-down list, select **= (equal to)**. In the **Value 1** field, type **Over**. Format the result to have **blue**, **bold**, **italic** text and a **yellow** fill color.



9. Click **OK** to return to the Alerters dialog box.

10. Apply the alerter to the cell where you typed the label in step 5. Your report should look similar to the one below.

2002

Quarter	State	Quantity sold
Q1	California	3,509
	Colorado	921
	DC	1,467
	Florida	924
	Illinois	1,711
	Massachusetts	609
	New York	3,717
	Texas	5,278
	Q1 Total:	**18,136**
	2002 Average:	**13,270**
Q1 Was Over Average for 2002		

Highlighting Table Blocks That Are Over a Report Average

Problem:

I want above average blocks in my report to stand out, but I want to calculate the average in the context of the entire report.

Solution:

11. Create a new report using **Year**, **Quarter**, **State**, and **Quantity sold** from the **eFashion** universe.

12. Section the report by **Year** and apply a break on **Quarter**.

13. Enter the following formula in the break footer to display the average of the combination of **Year** and **Quarter** within the report:

```
=Average(<Quantity sold> ForAll <State>) In Report
```

14. Next, you will need a variable that determines whether the **Quarter** total is above or below the average for the report. Create a new variable named **Over or Under Report Average** with the following formula:

```
=If(Sum(<Quantity sold>) - Average(<Quantity sold> ForAll <State>)
In Report) > 0 Then "Over" Else "Under"
```

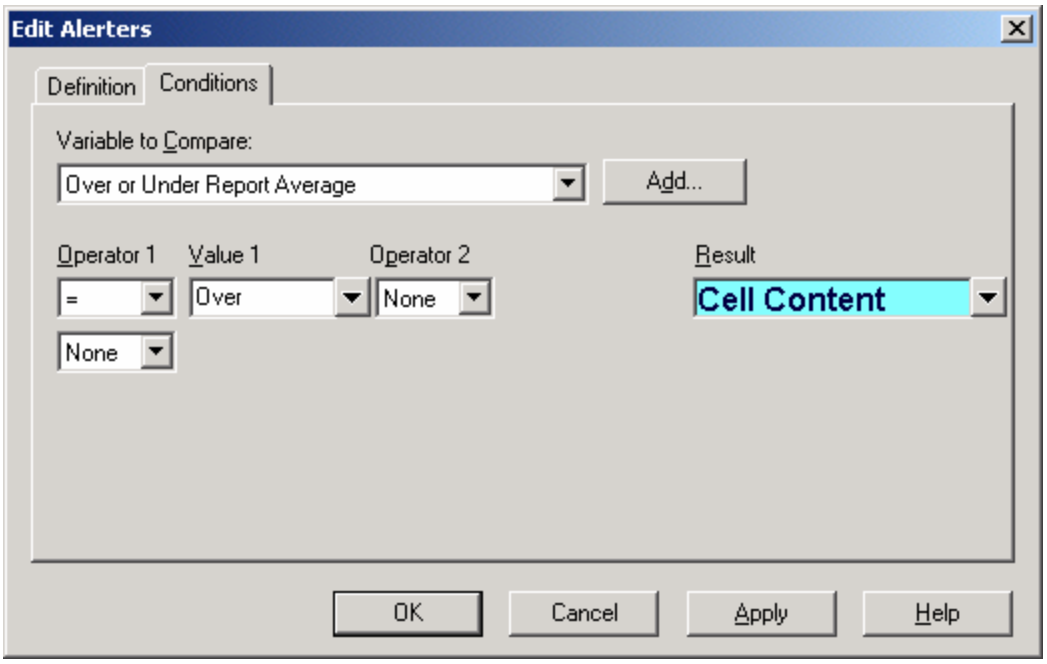
15. Next, create a label in the break footer that concatenates the result of the **Over or Under Report Average** variable with some descriptive text:

```
=<Quarter> + " Was " + <Over or Under Average> + " Average for the
Report"
```

16. Now that the average has been calculated, the variable that displays "Over" or "Under" has been created, and the label has been entered, you can apply an alerter to the label to make it stand out when the total for the **Quarter** is above the average for the **Year**. To do this, create a new alerter named **Highlight Over Report Average**.

17. On the **Conditions** tab, select **Over or Under Report Average** from the Variable to Compare drop-down list.

18. From the **Operator 1** drop-down list, select **= (equal to)**. In the **Value 1** field, type **Over**. Format the result to have **blue**, **bold**, **italic** text and a **yellow** fill color.



19. Click **OK** to return to the Alerters dialog box.

20. Apply the alerter to the cell where you typed the label in step 5. Your report should look similar to the one below.

2003

Quarter	State	Quantity sold
Q1	California	4,145
	Colorado	1,178
	DC	1,812
	Florida	1,095
	Illinois	2,079
	New York	4,248
	Texas	6,578
	Q1 Total:	**21,135**
	2003 Average:	**19,964**
	Average for Report:	**18,602**
Q1 Was Over Average for the Report		