

Is My Essbase Calculation Seeing Deja Vu All Over Again?

Everybody knows the quickest way from point A to point B is a straight line. Everybody assumes that the path is traveled only one time – not back and forth, over and over again. I see a lot of Essbase calculations and business rules, from experienced and novice developers, that go from point A to point B taking a straight line. But, the calculation travels that line multiple times and is terribly inefficient.

Here is a simple example of a calculation. Assume the Account dimension is dense, and the following members are all members in the Account dimension. We will also assume there is a reason to store these values rather than making them dynamic calc member formulas. Most of these are embedded in a FIX statement so the calculation only executes on the appropriate blocks. To minimize confusion, this will not be added to the example.

```
Average Balance = (Beginning Balance Ending Balance) / 2;  
Average Headcount = (Beginning Headcount Ending Headcount) /  
2;  
Salaries = Average Headcount * Average Salaries;  
Taxes = Gross Income * Tax Rate;
```

One of the staples of writing an effective calculation is to minimize the number of times a single block is opened, updated, and closed. Think of a block as a spreadsheet, with accounts in the rows, and the periods in the columns. If 100 spreadsheets had to be updated, the most efficient way to update them would be to open one, update the four accounts above, then save and close the spreadsheet (rather than opening/editing/closing each spreadsheet 4 different times for each account).

I will preface by stating the following can respond differently in different version. The 11.1.x admin guide specifically states the following is not accurate. Due to the inconsistencies I have experienced, I always play it safe and assume the following regardless of the version.

You might be surprised to know that the example above passes through every block four times. First, it will pass through all the blocks and calculate Average Balance. It will then go back and pass through the same blocks again, calculating Average Headcount. This will occur two more times for Salaries and Taxes. This is, theoretically, almost 4 times slower than passing through the blocks once.

The solution is very simple. Simply place parenthesis around the calculations.

```
(  
Average Balance = (Beginning Balance Ending Balance) / 2;  
Average Headcount = (Beginning Headcount Ending Headcount) /  
2;  
Salaries = Average Headcount * Average Salaries;  
Taxes = Gross Income * Tax Rate;  
)
```

This will force all four accounts to be calculated at the same time. The block will be opened, all four accounts will be calculated and the block will be saved.

If you are new to this concept, you probably have done this without even knowing you were doing it. When an IF statement is written, what follows the anchor? An open parenthesis. And, the ENDIF is followed by a close parenthesis. There is your block!

```
"East"  
(IF(@ISMBR("East"))  
    "East" = "East" * 1.1;  
ENDIF)
```

I have seen this very simple change drastically improve calculations. Go back to a calculation that can use blocks and test it. I bet you will be very pleased with the improvement.