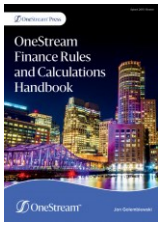
OneStream™ Press



OneStream Finance Rules and Calculations Handbook

Errata

Chapter 3, page 59

Original:

Given the above required syntax, only the denominator can have the unbalanced Dimensions attached to it. What do we do when the numerator contains the unbalanced Dimensions?

The answer is to use some basic third grade math and convert division into multiplication. We can divide the numerator by 1 and use `MultiplyUnbalanced` instead, since the order of multiplication does not matter.

```
  MultiplyUnbalanced(DenominatorMemberScript, Divide(1,
NumeratorMemberScript), UnbalancedScript)
```

Revised:

Given the above required syntax, only the denominator can have the unbalanced Dimensions attached to it. What do we do when the numerator contains the unbalanced Dimensions?

The answer is to use some basic third grade math and convert division into multiplication. **We can divide 1 by the numerator** and use `MultiplyUnbalanced` instead, since the order of multiplication does not matter.

*

Chapter 3, page 62

Original:

The below script shows how to use this function to convert a Data Buffer from the **Budget Scenario** to the dimensionality of the **Actual Scenario** by collapsing the detail in the Budget to the common Member

Revised:

The below script shows how to use this function to convert a Data Buffer from the **Actual Scenario** to the dimensionality of the **Budget Scenario** by collapsing the detail in the Actual to the common Member.

*

[page 66 states] Use Intelli-Sense to ensure each Dimension is in the right argument position. For example, `UD1` needs to always be in the fifth position as shown above.

Therefore, in the top row of Figure 3.37 [**Chapter 3, page 75**] should utilize five commas and not three.

Original:

api.Data.ClearCAlculatedData(True,True, True, True, “A#Sales”,,,”U1#Insects.Base”)

Revised:

api.Data.ClearCAlculatedData(True,True, True, True, “A#Sales”,,,,,”U1#Insects.Base”)