



# A Calculated Risk

*A detailed investigation into the complexities and limitations of XBRL Calculations*

By:

Shawn Rush  
CompSci Resources, LLC





**Table of Contents**

Introduction ..... 4

Basics..... 4

Complexities and Limitations..... 7

    Period Type ..... 7

    Basic Accounting Equation..... 8

    Dimensional Calculations..... 8

    Income and Cash Flows Statement..... 9

    Calculation Bleed-through ..... 11

    Negation Abuse..... 13

Conclusion..... 14

About the Author ..... 15

## Introduction

Few individuals in the eXtensible Business Reporting Language (XBRL) community would disagree that calculations are one of the more confusing and frustrating aspects of the specification. In theory, calculations should be fairly straightforward. All the numbers on the financial statements should add up to the stated totals, but as we know the devil is in the details. Hopefully we can shed some light on those details by reviewing not only the basics of creating calculations, but also certain scenarios that are either tricky or downright impossible given the current limitations of the specification.

## Basics

We'll start with the basics of how XBRL calculations work. Let's say we have the following on our balance sheet:

Label	2012	2011	XBRL Concept
Total liabilities	110	100	Liabilities
Total shareholder's equity	120	90	StockholdersEquity
Total liabilities and shareholder's equity	230	190	LiabilitiesAndStockholdersEquity

In our calculation linkbase, the XML tags for this calculation would look something like this:

**<calculationArc**

```

xlink:type="arc"
weight="1"
order="1"
xlink:to="us-gaap_Liabilities"
xlink:from="us-gaap_LiabilitiesAndStockholdersEquity"
xlink:arcrole="http://www.xbrl.org/2003/arcrole/summation-item"/>
    
```

**<calculationArc**

```

xlink:type="arc"
weight="1"
order="2"
xlink:to="us-gaap_StockholdersEquity"
xlink:from="us-gaap_LiabilitiesAndStockholdersEquity"
xlink:arcrole="http://www.xbrl.org/2003/arcrole/summation-item"/>
    
```

Note that “xlink:to” denotes the operand whereas “xlink:from” denotes the total. The other important item to look at is the “weight” value. While the XBRL specification allows any numeric here, the U.S. Securities and Exchange Commission (SEC) only allows values of either [1] or [-1]. Weight determines whether we are adding or subtracting the operand. Hence, the above XBRL is expressed in a human-readable format by the following equation:

$$\text{Liabilities} + \text{StockholdersEquity} = \text{LiabilitiesAndStockholdersEquity}$$

Note that only concepts are specified in a calculation. Fact values, contexts (which carry date and dimension information), and presentation links (tables or sections) are not specified when creating the calculation. The calculation applies to a set of facts for a given context if (a) a fact exists for the total concept with the given context and (b) another fact exists for at least one of the operand concepts with the given context. Again, context specifies the dates and dimensions for a given fact value.

Let’s take a look at how this applies if we have the following data:

Label	2012	2011	2010	2009	2008	XBRL Concept
Total liabilities	110	100	N/A	N/A	N/A	Liabilities
Total shareholder’s equity	120	90	N/A	80	70	StockholdersEquity
Total liabilities and shareholder’s equity	230	190	180	N/A	160	LiabilitiesAndStockholdersEquity

*N/A indicates that no fact was reported*

Given the scenario above, the table below shows how the calculation is applied to the data:

Context	Liabilities		StockholdersEquity		LiabilitiesAndStockholdersEquity	Does Calculation Apply?
<b>2012</b>	110	+	120	=	230	Yes
<b>2011</b>	100	+	90	=	190	Yes
<b>2010</b>	N/A	+	N/A	=	180	No (needs at least 1 operand)
<b>2009</b>	N/A	+	80	=	N/A	No (needs total fact)
<b>2008</b>	N/A	+	70	=	160	Yes (but invalid)

With this we must remember that the calculations are applied to facts grouped solely by context and not by financial statement or presentation link. This means that a single calculation can apply to multiple presentation links if those links contain concepts that exist in the calculation. We will discuss some of the issues with this later.

A concept's balance type also plays a significant role in calculations. A concept could have a balance type of debit, credit, or none. This is relevant because the balance type of the total concept determines whether the operands can be added or subtracted.

If this sounds confusing to you, then you're not alone. We'll use the previous calculation to shed some light.

<b>Type</b>	<b>Concept</b>	<b>Balance Type</b>
<i>Operand 1</i>	+ Liabilities	<i>Debit</i>
<i>Operand 2</i>	+ StockholdersEquity	<i>Debit</i>
<i>Total</i>	LiabilitiesAndStockholdersEquity	<i>Debit</i>

Because "LiabilitiesAndStockholdersEquity" has a balance type of debit and is the total, debit concepts can only be added and credit concepts can only be subtracted. Likewise, if the total were to have a balance type of credit, credit concepts can only be added and debit concepts can only be subtracted. Note that if a total does not have a balance type, then credit concepts and debit concepts can be either added or subtracted. Similarly, if an operand concept has no balance type, it can be either subtracted or added regardless of the balance type of the total.

In the actual calculation in the XBRL, the addition or subtraction of a concept is denoted by a calculation weight of either [1] or [-1], respectively. Note, however, that this rule regarding balance types and valid calculations weights does not put any restriction on whether the fact value itself is positive or negative. Rules regarding the sign of a fact value are inherent to the concept definition and are not determined by the balance type. For example, "ProfitLoss" can be either positive or negative, but "CommonStockSharesOutstanding" cannot be negative. This is important to remember as one could incorrectly adjust the sign of a fact value to overcome calculation weight restrictions based on balance type. Such action makes the calculation "work", but renders the data meaningless. We will show an example of this later.

The following are valid calculations based on balance type:

<b>Operand 1</b>		<b>Operand 2</b>		<b>Total</b>	
+	Credit	+	Credit	=	Credit
+	Credit	-	Debit	=	Credit
+	Debit	+	Debit	=	Debit
-	Debit	-	Debit	=	Credit
-	Credit	-	Credit	=	Debit
+	Credit	+	Debit	=	No Balance Type
+	Credit	-	No Balance Type	=	Credit

The following are invalid calculations based on balance type:

<b>Operand 1</b>		<b>Operand 2</b>		<b>Total</b>	<b>Reason Invalid</b>
+	Credit	-	Credit	= Credit	Operand 2 weight must be +
+	Credit	+	Debit	= Credit	Operand 2 weight must be -
+	Debit	-	Debit	= Debit	Operand 2 weight must be +
+	Debit	-	Debit	= Credit	Operand 1 weight must be -
-	Credit	+	Credit	= Debit	Operand 2 weight must be -
-	Credit	-	No Balance Type	= Credit	Operand 1 weight must be +

Finally, a concept's period type plays a role in how we can set up calculations. A single calculation can only contain concepts of the same period type. This means instants can only add up to instants and durations can only add up to durations.

## Complexities and Limitations

### *Period Type*

One of the more noticeable limitations of the calculation specification is its restriction on mixing concepts of different period types in a single calculation. This limitation becomes obvious in the inability to specify a calculation in the stockholder's equity statement. The general structure of an equity table shows the stockholder's equity at a given point in time (instant) followed by changes over the course of some period of time (duration) whether it be a year or a quarter followed by the new balance (instant).

<b>Label</b>	<b>Value</b>	<b>Period Type</b>
Balance as of 12/31/2011	100	Instant
Net Income	10	Duration
Other Comprehensive Income	20	Duration
Dividends Paid	20	Duration
Balance as of 12/31/2012	150	Instant

Despite the calculation being obvious in the table, we are unable to model it using XBRL calculations.

### *Basic Accounting Equation*

The basic accounting equation using XBRL concepts is as follows:

$$\text{Assets} = \text{Liabilities} + \text{StockholdersEquity}$$

Unfortunately, this simple equation cannot be modeled with XBRL calculations because of balance type restrictions. Given that “Assets” is a debit and “Liabilities” and “StockholdersEquity” are both credits, there is no way to write this equation in XBRL with the proper calculation weights. In the equation above, either “Assets” or both “Liabilities” and “StockholdersEquity” would have to have a negative sign. Neither scenario makes sense so this important calculation cannot be included in the calculation linkbase.

### *Dimensional Calculations*

As mentioned before, calculations only apply to facts that share the same context. Hence, if two facts have different dimensions (and consequently different contexts), then they cannot be part of the same calculation.

Consider the example below:

<b>Concept</b>	<b>Dimension</b>	<b>Value</b>
CommonStockSharesOutstanding	CommonClassAMember	2,000,000
CommonStockSharesOutstanding	CommonClassBMember	1,000,000
CommonStockSharesOutstanding	[None]	3,000,000



We have three facts all of which have the concept “CommonStockSharesOutstanding”. The dimensionless fact represents the total and the other two facts with dimensions represent the components or, in terms of a calculation, the operands. This is a quintessential application of dimensions – to break a data point into its components; however, there is no way to model it using XBRL calculations given the context restrictions.

### *Income and Cash Flows Statement*

The relationship between the income statement and cash flows statement poses some interesting challenges regarding calculations. In the income statement, we show a number of items that ultimately add up to the total, “ProfitLoss”. When creating the total for “NetCashProvidedByUsedInOperatingActivities” in the statement of cash flows, there are two options: the direct method and indirect method. To arrive at this total, the direct method adds and subtracts the cash items, whereas the indirect method starts with “ProfitLoss” and makes adjustments for non-cash items. As we will investigate below, the indirect method poses certain challenges when tagging XBRL.

Let’s consider the following simplified income statement:

<b>Concept</b>	<b>Value</b>	<b>Balance Type</b>
Revenues	10,000	Credit
DepreciationDepletionAndAmortization	1000	Debit
AssetImpairmentCharges	500	Debit
ProfitLoss	8,500	Credit

We can create the following simple calculation for “ProfitLoss” based on the balance types:

$$\text{Revenues} - \text{DepreciationDepletionAndAmortization} - \text{AssetImpairmentCharges} = \text{ProfitLoss}$$

Everything looks good so far. Now let us look at the corresponding cash flows statement:

<b>Concept</b>	<b>Value</b>	<b>Balance Type</b>
ProfitLoss	10,000	Credit
DepreciationDepletionAndAmortization	1,000	Debit
NetCashProvidedByUsedInOperatingActivities	11,000	No Balance Type

The first thing to notice is that “NetCashProvidedByUsedInOperatingActivities” lacks a balance type, which may seem odd at first. Given that this is a component of “Cash and Cash Equivalents,” one might expect this concept to be a debit. Additionally, both “NetCashProvidedByUsedInFinancingActivities” and “NetCashProvidedByUsedInInvestingActivities” are debits.

If, however, “NetCashProvidedByUsedInOperatingActivities” were a debit, we would not be able to create the following calculation:

**ProfitLoss + DepreciationDepletionAndAmortization = NetCashProvidedByUsedInOperatingActivities**

If “NetCashProvidedByUsedInOperatingActivities” were a debit, then we would be forced to subtract “ProfitLoss” which does not make sense here. Fortunately, “NetCashProvidedByUsedInOperatingActivities” does not have a balance type so we are free to create the above calculation. Additionally, a portion of the definition for this concept appears to support this reason for no balance type:

*While for technical reasons this element has no balance attribute, the default assumption is a debit balance consistent with its label.*

Another issue that causes confusion with the cash flows statement is a calculation that exists in the income statement but in reverse fashion. Consider the following example from the income statement:

<b>Concept</b>	<b>Value</b>	<b>Balance Type</b>
IncomeLossFromContinuingOperations	10,000	Credit
IncomeLossFromDiscontinuedOperationsNetOfTax	5,000	Credit
DiscontinuedOperationGainLossOnDisposalOfDiscontinuedOperationNetOfTax	1,000	Credit
NetIncomeLossAvailableToCommonStockholdersBasic	16,000	Credit

Sometimes you could see the exact same items in the cash flows statement but in reverse order such as the following:

<b>Concept</b>	<b>Value</b>	<b>Balance Type</b>
NetIncomeLossAvailableToCommonStockholdersBasic	16,000	Credit
DiscontinuedOperationGainLossOnDisposalOfDiscontinuedOperationNetOfTax	1,000	Credit
IncomeLossFromDiscontinuedOperationsNetOfTax	5,000	Credit
IncomeLossFromContinuingOperations	10,000	Credit

Creating a calculation for the income statement is not a problem, but we cannot create the calculation in the exact form that is presented in the cash flows statement. From the income statement, we have the following equation:

$$\text{IncomeLossFromContinuingOperations} + \text{IncomeLossFromDiscontinuedOperationsNetOfTax} + \text{DiscontinuedOperationGainLossOnDisposalOfDiscontinuedOperationNetOfTax} = \text{NetIncomeLossAvailableToCommonStockholdersBasic}$$

Algebra tells us that we can rewrite this equation as follows:

$$\text{IncomeLossFromContinuingOperations} = \text{NetIncomeLossAvailableToCommonStockholdersBasic} - \text{DiscontinuedOperationGainLossOnDisposalOfDiscontinuedOperationNetOfTax} - \text{IncomeLossFromDiscontinuedOperationsNetOfTax}$$

From an XBRL perspective, since every concept in these equations is a credit including the total, the first equation is fine. We cannot, however, rewrite the first equation as the second in XBRL because we cannot have a negative calculation weight on the last two operands since those operands and the total are credits. Now given that the two calculations are logically equivalent, it is sufficient to just include the first one in the XBRL for both. In fact, the first calculation is valid for the cash flows statement even if that same data did not exist in the income statement.

*Calculation Bleed-through*

One of the most common issues filers encounter deals with creating a calculation for a set of concepts for one table and context and the calculation inadvertently applying to a very similar but different set of concepts for a different context. Consider the following subsection of an income statement:

Concept	Value	Balance Type
Revenues	10,000	Credit
OperatingExpenses	6,000	Debit
OperatingIncomeLoss	4,000	Credit

From this we have the following calculation:

$$\text{Revenues} - \text{OperatingExpenses} = \text{OperatingIncomeLoss}$$

Now let's consider a table in the notes that breaks down expenses based on industry segments via dimensions:

<b>Concept</b>	<b>Hotels</b>	<b>Corporate</b>	<b>Consolidated</b>	<b>Balance Type</b>
Revenues	10,000	0	10,000	Credit
DirectCostsOfOwnedHotels	1,000	0	1,000	Debit
GeneralAndAdministrativeExpense	1,500	1,500	3,000	Debit
DepreciationExpense	2,000	0	2,000	Debit
OperatingIncomeLoss	5,500	(1,500)	4,000	Credit

In this table, the Hotels and Corporate columns would be tagged with separate Business Segment dimensions, whereas the Consolidated column would be dimensionless since it is the aggregate. The amounts 10,000 and 4,000 in the Consolidated column would already have been tagged from the income statement.

The issue here arises when we now try to apply the calculation from the first table to the facts in the second table. Remember when applying calculations, we do so by context which means the calculation from the first table gets applied to each of the three columns of the second table. When we try to apply the calculation to the first two columns, there is no "OperatingExpenses" fact for either of those contexts. This causes the calculation to be applied as follows:

$$\text{Revenues} = \text{OperatingIncomeLoss}$$

This is obviously wrong and will flag a calculation inconsistency for those first two columns or contexts. In these circumstances, the SEC requires that the calculation for the income statement be included as it is valid when applied to the "Required Context". The "Required Context" is the main dimensionless context of the document whose dates match the reporting period of the filing.

One thing to remember is that the calculation will apply correctly to the Consolidated column. The facts for "Revenues" and "OperatingIncomeLoss" in this column are the exact same facts as those in the income statement as they share the same dimensionless context. This means the application of the calculation to the Consolidated column is one and the same as the application of the calculation to the income statement.

Note that this overall scenario could apply to differences in context due to dates as well. For example, if the second table showed a Consolidated column for a date range different than the one reported in the

income statement, then we would encounter a similar calculation inconsistency for the Consolidated column.

This type of scenario is one of the main causes of confusion regarding the validity of an XBRL submission. Many validation tools will understandably flag this scenario as a calculation inconsistency even though the filer is following the SEC's instructions and has created the calculation properly. Because validation tools have not been able to differentiate between this scenario and a real calculation error, there is a general misconception that XBRL data contains more calculation errors than actually exist. Addressing this issue will be critical to the overall effort of increasing the consumption of XBRL data.

### *Negation Abuse*

The final issue discussed here deals with applying negated label roles incorrectly to force valid calculations while still having the rendered value match the value in the HTML. Consider the following table and calculation:

<b>Concept</b>	<b>Label</b>	<b>Value</b>	<b>Label Role</b>	<b>Rendered Value</b>	<b>Balance Type</b>
Revenues	Revenues	10,000	Standard	10,000	Credit
OperatingExpenses	Operating Expenses	6,000	Standard	6,000	Debit
OperatingIncomeLoss	Operating Income	4,000	Total	4,000	Credit

$$\text{Revenues} - \text{OperatingExpenses} = \text{OperatingIncomeLoss}$$

Note that "Operating Expenses" is expressed as a positive number. Sometimes in the HTML, filers may want to express this with parentheses so that the line items can all be summed. In this case, the proper structure would be as follows:

<b>Concept</b>	<b>Label</b>	<b>Value</b>	<b>Label Role</b>	<b>Rendered Value</b>	<b>Balance Type</b>
Revenues	Revenues	10,000	Standard	10,000	Credit
OperatingExpenses	Operating Expense	6,000	Negated	(6,000)	Debit
OperatingIncomeLoss	Operating Income	4,000	Total	4,000	Credit

Note that fact value of 6,000 in both examples is the same, but in the second example, we have applied the Negated label role so that a negative value is rendered to match the HTML. The calculation above still works for this setup as the fact values did not change.

Now consider the following:

Concept	Label	Value	Label Role	Rendered Value	Balance Type
Revenues	Revenues	-10,000	Negated	10,000	Credit
OperatingExpenses	Operating Expense	-6,000	Standard	(6,000)	Debit
OperatingIncomeLoss	Operating Income	-4,000	NegatedTotal	4,000	Credit

From a rendering perspective, this table matches the previous one, but the stored XBRL values are the complete opposite in sign. This is an error because the intention is to report positive revenues and expenses, but instead negative values were reported. Unfortunately, this could easily be missed because the rendering appears correct, and the calculation is still valid with all negative numbers.

Filers should be very careful to avoid this mistake by ensuring the reported value matches the intention of the concept. This requires filers to have a clear understanding of the concepts used and not to change the signs of fact values for the sole purpose of making calculations work.

## Conclusion

As we have shown, there are many considerations when applying XBRL calculations. The rules governing their use can appear complicated and overwhelming. There are limitations in the specification that add further confusion and can leave filers unsure as to whether or not they are applying calculations correctly. Many of the concerns can be addressed by a better understanding of the rules governing XBRL calculations especially with respect to contexts. As validation tools improve and the filer community becomes more educated in the nuances of the specification, calculations can shift from being a source of frustration to a source of confidence in the validity of XBRL data.

## About the Author

Shawn Rush is a Senior Software Engineer at CompSci Resources, LLC. He earned a degree in Systems Engineering from the University of Virginia and has extensive experience consulting a wide array of clients on the XBRL specification and developing XBRL-related software products. Mr. Rush is a co-inventor of CompSci's patented XBRL technology that automates the review, organization, tagging, and analysis of data for generating XBRL documents.

You can connect with Mr. Rush on LinkedIn by following: [www.linkedin.com/profile/view?id=64679081](http://www.linkedin.com/profile/view?id=64679081). CompSci's LinkedIn page can be found here: [www.linkedin.com/company/compsci-resources-llc](http://www.linkedin.com/company/compsci-resources-llc), and you can learn more about the company on its website: [www.compsciresources.com](http://www.compsciresources.com).

