Data View Web Part Code

It's easy to write a piece on using DataViews that shows how easy it is to create good-looking alternative presentations of a list by using only menu-guided methods. It's another thing to easily satisfy real-world requests and, as most of us know, bosses want to be warned of problems in advance.

So, although I included the necessary steps for create a good-looking data view in Hour 21, the main section of that hour shows how to take a real-world request and solve it using a Data View Web Part. While it was possible to create a data view that matched most of that request by using only menu-guided methods, there was one item—getting a sum of staff numbers per company—that was not quickly achieved. So, I ended that section in the book with the warning not to assume that every request would be possible to achieve without a lot of work or some code.

This website section picks up where the section of the book left off.

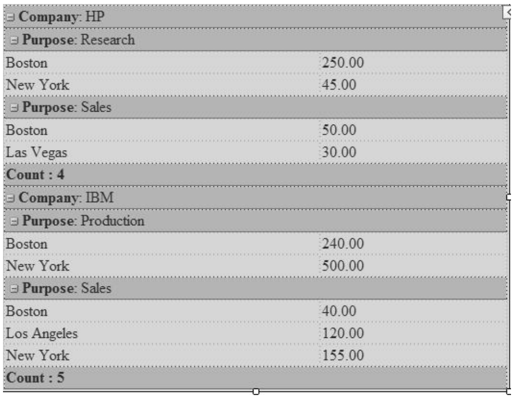
It starts with Figure 1, which shows that we have Count=4 (number of rows) rather than the sum of staff numbers per company, which is what we want to see. It then continues with several attempts to solve this problem. Each of these attempts gets closer to solving the problem, but in the end it has to be conceded that the problem needs code to finally solve it and that code is provided.

So be warned. In order to be sure that you can handle real-world requests, you need to have knowledge of XPath and XSLT.

There are plenty of books (especially on XSLT) that define these languages, but very few (particularly not for XPath) provide sample examples. I know this only too well because I spent many hours staring at Internet pages and a detailed book on XPath and XSLT and still, as you will see, only got very close to the solution. I then asked a group of SharePoint MVPs what I was missing and a final hint from one of them solved the problem.

I hope you enjoy this online section. I suggest you work through it rather than just jumping to the end for the solution because I think you'll learn a lot on the way.

Select the row containing Count: 4 and right-click. Figure 2 shows part of what you then see. I've inserted an arrow indicating what you then need to select.



⊟ **Company**: HP	
⊟ **Purpose**: Research	
Boston	250.00
New York	45.00
⊟ **Purpose**: Sales	
Boston	50.00
Las Vegas	30.00
Count : 4	
⊟ **Company**: IBM	
⊟ **Purpose**: Production	
Boston	240.00
New York	500.00
⊟ **Purpose**: Sales	
Boston	40.00
Los Angeles	120.00
New York	155.00
Count : 5	

FIGURE 1
No repeated names.

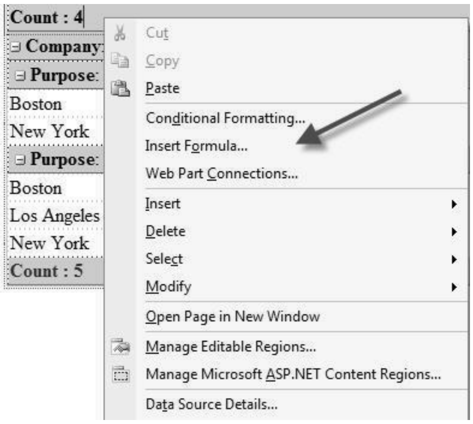
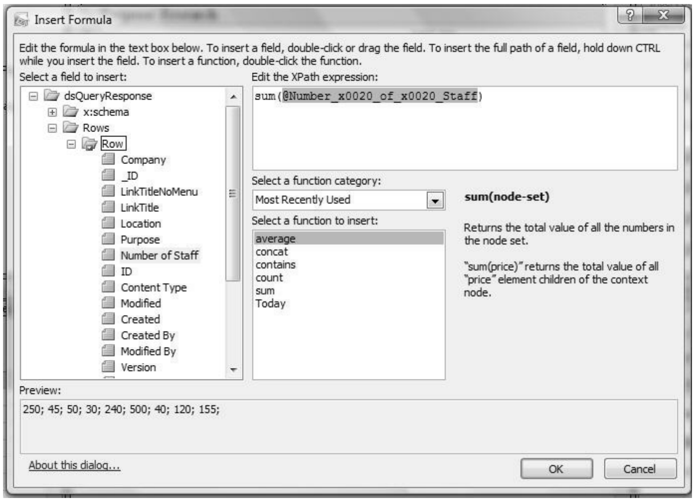


FIGURE 2
Inserting a formula.

First, double-click Sum, and then double-click Number of Staff. You should see Figure 3.



Insert Formula

Edit the formula in the text box below. To insert a field, double-click or drag the field. To insert the full path of a field, hold down CTRL while you insert the field. To insert a function, double-click the function.

Select a field to insert:

- dsQueryResponse
  - x:schema
  - Rows
    - Row
      - Company
      - _ID
      - LinkTitleNoMenu
      - LinkTitle
      - Location
      - Purpose
      - Number of Staff
      - ID
      - Content Type
      - Modified
      - Created
      - Created By
      - Modified By
      - Version

Edit the XPath expression:

```
sum(@Number_x0020_of_x0020_Staff)
```

Select a function category: Most Recently Used

Select a function to insert:

- average
- concat
- contains
- count
- sum
- Today

sum(node-set)

Returns the total value of all the numbers in the node set.

"sum(price)" returns the total value of all "price" element children of the context node.

Preview:

```
250; 45; 50; 30; 240; 500; 40; 120; 155;
```

About this dialog...

OK Cancel

FIGURE 3
The first attempt at a good sum formula.

I think you'll agree that this looks to be a reasonable solution. After all, we are on the footer line for the first grouping level, so we can reasonably expect our sum to be at that level.

But click OK, and you have Count: 4240 and Count: 5155, respectively, for HP and IBM when you should be getting Total Number of Staff: 375 and Total Number of Staff: 1055. The text part is easy enough to fix. Just like adding bold to HP (which we did in the book), go to the Count text and replace it with Total Number of Staff. But, where are these numbers coming from?

A look at the generated code gives the answer. (When in Design mode in SharePoint Designer 2007, place your cursor at the end of the 4240 and then select the Code mode [or Split].):

Number of Staff: `<xsl:value-of select="count($nodeset)" /><xsl:value-of select="sum(@Number_x0020_of_x0020_Staff)" />`

When you insert the correct formula, it is added on top of the existing formula rather than replacing it.

Note

The difference between XPath and XSLT is as follows: `[sum(@Number_x0020_of_x0020_Staff) ]` was XPath code, and what was generated from that `[<xsl:value-of select="sum(@Number_x0020_of_x0020_Staff)" />]` is XSLT code.

From that code, we get rid of `<xsl:value-of select="count($nodeset)" />`, and we now have Total Number of Staff: 240 and Total Number of Staff: 155.

These are the values for what exactly?

Well, they seem to be values of the first and last lines for IBM and no sum at all. What we do know is that we need to look at how the correct count was achieved and then try to work out how to get the correct sum. To do that, we need to find out what a nodeset is and transfer that idea into the Number of Staff field.

Note

I know the word nodeset because after many hours of staring (and mentally screaming) at Xpath/XSLT texts, I knew that the sum function requires a nodeset rather than a field. I also knew that the only examples I could find said that to make a field a nodeset you need to add // before the field name.

So let's try:

```
sum(//@Number_x0020_of_x0020_Staff)
```

Figure 4 is the result (together with a suitably revised text), which is the correct total staff count for all the companies. However, this total staff count appears everywhere you would expect a total staff count **per company**. So it's still not quite right.

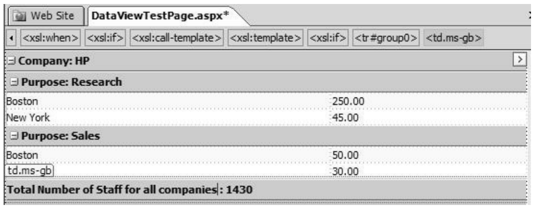


FIGURE 4
Almost achieving the total number of staff per company.

Web Site
DataViewTestPage.aspx*
<xsl:when> <xsl:if> <xsl:call-template> <xsl:template> <xsl:if> <tr#group0> <td.ms-gb>
td.ms-gb

Company: HP

Purpose: Research

Boston	250.00
New York	45.00

Purpose: Sales

Boston	50.00
	30.00

Total Number of Staff for all companies: 1430

Rather than attacking those XPath/XSLT texts again, this is where I requested help from other SharePoint MVPs, and MVP Jeremy Sublett provided me with this (XSLT) formula:

```
<xsl:value-of select="sum($nodeset/@NumberOfStaff)" />
```

I'll leave it to you to work out what information you choose to enter into Figure 3 to get this in your code (remember, what you enter there will be in XPath format). Hint: Look at the Note after Figure 3 and at the formula used to get Figure 4.