Part 2 - Working with Formulas

Table of Contents

- Intro
- Working with Formulas
 - Combining Data Markers with Text
 - Using Inline Formulas
 - Adding a Total Row
- Final Code
- Downloads
- Next Steps

Intro



This is Part 2 of a 3-part tutorial series for the Simple Expense Summary scenario. It is recommended that you complete Part 1 - Getting Started before starting this section.



Following the Sample

There is a downloadable ExcelWriter Basic Tutorials.zip with completed templates and code. The completed example of the template is available under templates/part2_template.xlsx. The code for this part of the tutorial can be found in Part2.aspx.cs.

This part focuses on adding some Excel formulas to the template file from Part 1. Specifically, this covers combining data marker values with other text, adding a 'Total' row after imported data, and including formulas in imported data rows.

Since the formulas are native Excel functionality, we will only be modifying the template file. There are no changes to the code from Part 1.

Working with Formulas

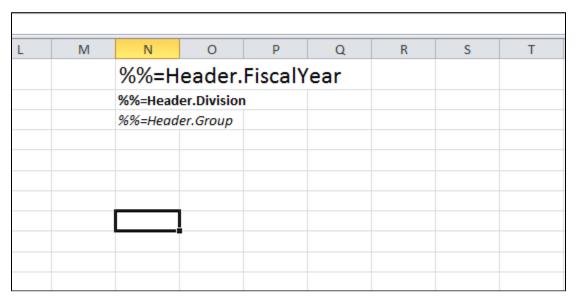
Combining Data Markers with Text

We start with the template file as it was at the end of Part 1:

	F16 ▼ (*) × ✓ f _x								
4	Α	В	С	D	Е	F	G		
1	%	%=Header.FiscalYear							
2		%%=Header.Division							
3		%%=Header.Group							
4									
5		Top Expenses							
6		Description	Expenses						
7		%%=[Top 5 Expenses].Description	%%=[Top 5 Expenses].	Expenses					
8									
9		All Expenses							
10		Description	Expenses						
11		%%=[All Expenses].Description	%%=[All Expenses].Exp	oenses					
12									
13									
14									
15									
16									
17									
18									
19									
20									

Right now, the header is set up just to display the values for FiscalYear, Division, and Group. Let's say we want to change the labels to "Expenses Summary - <Fiscal Year>", "Division: <Division>", "Group: <Group>". Data markers must be the only content a cell, they cannot be combined with other text. To get around this, you need to build the text or formula with a reference to the cell the data marker occupies.

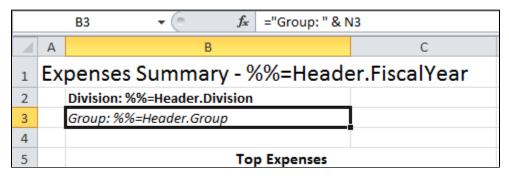
1. Move the data markers for %%=Header.FiscalYear, %%=Header.Division, %%=Header.Group to a column off to the right, say Column N



2. In A1, replace %%=Header.FiscalYear with a formula that builds the desired string with the formula = "Expenses Summary - " & N1.

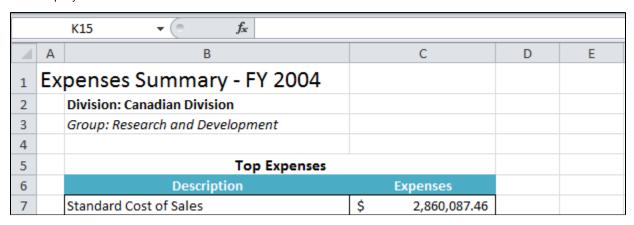
	A1 • fx ="Expenses Summary - " & N1						
	Α	В	С	D			
1	Ex	penses Summary - %%=Heade	er.FiscalYear				
2		%%=Header.Division					
3		%%=Header.Group					
4							
5		Top Expenses					
6		Description	Expenses				

3. In B2 and B3, replace %%=Header.Division and %%=Header.Group with similar formulas: ="Division: " & N2, ="Group: " & N3.



- 4. Hide column N so the extra data markers won't be visible.
- 5. Run the code.

In the output you will see the text combined with the values of the data markers:



Using Inline Formulas

If a formula references a data marker row, ExcelWriter will update the formula to reflect that multiple rows are being inserted. In this section, we'll briefly cover how ExcelWriter handles formulas that are on the data marker row.



For more information about how ExcelWriter imports data with the ExcelTemplate object, check out How ExcelWriter Inserts Rows. That article also discusses how importing data with ExcelTemplate affects formulas.

Let's say we want to generate some short labels for the Top 5 Expenses to use with the descriptions. One possibility is to add the labels to the data source (i.e. through the SQL query or modifying the data structure directly), but another way is to use Excel formulas.

Similar to the last section, we'll use the existing data markers to create a new string for the Top 5 Expenses table.

6. In Column D, create a header for the label. Update the 'Top Expenses' merged cell area to include column D.

	B17 ▼ (f _x)								
4	Α	В	С	D	Е	F			
1	Ex	penses Summary - %%=Heade							
2		Division: %%=Header.Division							
3		Group: %%=Header.Group							
4									
5		Top Expense	25						
6		Description	Expenses	ID					
7		%%=[Top 5 Expenses].Description	%%=[Top 5 Expenses].	Expenses					
8									
9		All Expenses							
10		Description	Expenses						
11		%%=[All Expenses].Description	%%=[All Expenses].Ex	penses					
12									
13									
14									
15									

7. In cell D7, add a formula =UPPER(LEFT(B7, 4)).

Excel's LEFT(text, N) function returns the first N characters from the text, starting from the left. In this case, the first 4 characters from cell B7. Then the UPPER function converts the characters to uppercase.

ExcelWriter will update this formula for each row of data that is added, so the formula in row 9 will read =UPPER(LEFT(B9, 4)).

8. Run the report.

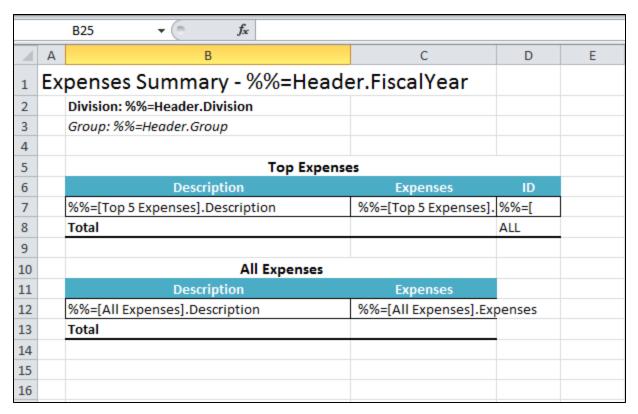
You will see that the first 4 characters from the description have been set to uppercase in column D.

	D9 \checkmark =UPPER(LEFT(B9, 4))								
	Α	В	С	D	Е	F	G		
1	Ex	penses Summary - FY 2004							
2		Division: Canadian Division							
3		Group: Research and Development							
4									
5		Top Expense	25						
6		Description	Expenses	ID					
7		Standard Cost of Sales	\$ 2,860,087.46	STAN					
8		Taxes	\$ 760,327.13	TAXE	ļ				
9		Salaries	\$ 641,302.20	SALA	ļ				
10		Variances	\$ 441,498.92	VARI	Ï				
11		Commissions	\$ 348,101.27	COMM					
12									

Adding a Total Row

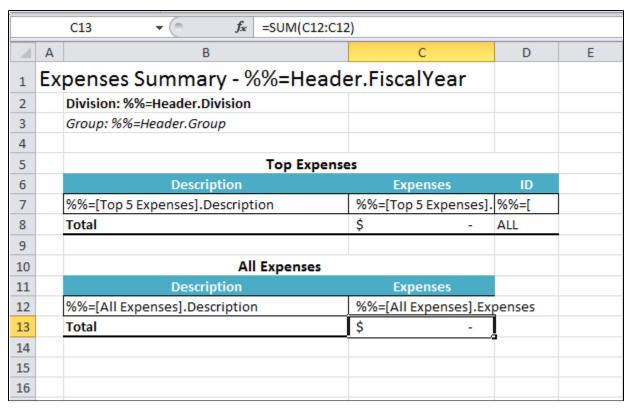
If an aggregate formula, such as TOTAL or SUM, references a data marker row, ExcelWriter will stretch the formula to include all the rows that have been inserted. In this section, we'll cover how to add a total row.

9. Insert a new row below Top Expenses and All Expenses. Add labels for the total row and format as desired.



^{10.} In cell C8, add the formula =SUM(C7:C7). Add a similar formula to C13: =SUM(C12:C12). Don't forget to format C8 and C13 to use currency formatting.

When ExcelWriter imports the data, the first formula will get stretched to C7:C11. The second formula will first get updated to account for the new data inserted above it (C16:C16) and then will get stretched to C16:C44.



11. Run the report.

You will see that the formulas have been stretched and updated as described above.

	C12						
4	Α	В		С	D	Е	
1	Ex	penses Summary - FY 2004					
2		Division: Canadian Division					
3		Group: Research and Development					
4							
5		Top Expense	25				
6		Description		Expenses	ID		
7		Standard Cost of Sales	\$	2,860,087.46	STAN		
8		Taxes	\$	760,327.13	TAXE		
9		Salaries	\$	641,302.20	SALA		
10		Variances	\$	441,498.92	VARI		
11		Commissions	\$	348,101.27	сомм		
12		Total	\$	5,051,316.98	ALL		
13				,			

	C45 ▼ (SUM(C16:C44)							
1	Α	В		С	D	Е		
28		Office Supplies	\$	5,652.48				
29		Other Assets	\$	2,546.37				
30		Other Expenses	\$	3,569.10				
31		Other Travel Related	\$	884.22				
32		Payroll Taxes	\$	65,023.29				
33		Professional Services	\$	6,681.27				
34		Rent	\$	12,013.18				
35		Returns and Adjustments	\$	274,069.61				
36		Salaries	\$	641,302.20				
37		Standard Cost of Sales	\$	2,860,087.46				
38		Taxes	\$	760,327.13				
39		Telephone	\$	90,696.48				
40		Travel Lodging	\$	7,941.90				
41		Travel Transportation	\$	8,734.34				
42		Utilities	\$	12,038.95				
43		Variances	\$	441,498.92				
44		Vehicles	\$	8,469.20				
45		Total	\$	5,803,082.23				
46								

Final Code



```
using SoftArtisans.OfficeWriter.ExcelWriter;
...
ExcelTemplate XLT = new ExcelTemplate();

XLT.Open(Page.MapPath("//templates//part1_template.xlsx"));

DataBindingProperties dataProps = XLT.CreateDataBindingProperties();

object[] valuesArray = { "FY 2004", "Canadian Division", "Research and Development" };

string[] columnNamesArray = { "FiscalYear", "Division", "Group" };

XLT.BindRowData(valuesArray, columnNamesArray, "Header", dataProps);

DataTable dtTop5 = GetCSVData(Page.MapPath("//data//Part1_Top5Expenses.csv"));

DataTable dtAll = GetCSVData(Page.MapPath("//data//Part1_AllExpenses.csv"));

XLT.BindData(dtTop5, "Top 5 Expenses", dataProps);

XLT.BindData(dtAll, "All Expenses", dataProps);

XLT.Process();

XLT.Process();

XLT.Save(Page.Response, "Part1_Output.xlsx", false);
```

Downloads

You can download the code for the Basic ExcelWriter Tutorials as a Visual Studio solution, which includes the Simple Expense Summary.

• ExcelWriter Basic Tutorials.zip

Next Steps

Continue on to Part 3: Adding a Chart