

Part 2 - Using Styles and Formatting

Table of Contents

- [Introduction](#)
- [Setting Up the Template](#)
 - [Number Formats](#)
 - [Conditional Formatting](#)
- [Adding an ExcelWriter Reference in Visual Studio](#)
- [Writing the Code](#)
 - [Data Binding](#)
- [Final Code](#)
- [Downloads](#)
- [Next Steps](#)

Introduction



This is Part 2 of the three-part tutorial series [Financial Report](#) scenario. It is recommended that you complete [Part 1 - Using Modifiers and Ordinal Syntax](#) before starting this section.

This tutorial also assumes a basic understanding of [how ExcelTemplate imports data](#) how that affects formulas and formatting.



Following the Sample Code

In the downloadable [ExcelWriter_Basic_Tutorials.zip](#), there is a completed template file located in *CompleteFinancialReport/templates/Part2_Financial_Template.xlsx*.

In this tutorial [ExcelTemplate](#) is being used to populate data in a template set up with styles and formats. This part of the tutorial uses [formulas](#) and persists Excel styles.

Setting Up the Template

The goal of this part of the tutorial is how to apply number formats and conditional formatting to ExcelTemplate files and what you can expect as ExcelTemplate imports data.

The final template will look like this:

	A	B	C	D	E	F	G
1							
2		Percent Change By Quarter					
3		Type	Q1-Q2	Q2-Q3	Q3-Q4		
4		Assets	#DIV/0!	#DIV/0!	#DIV/0!		
5		Losses	#DIV/0!	#DIV/0!	#DIV/0!		
6		Other	#DIV/0!	#DIV/0!	#DIV/0!		
7							
8		Details					
9		List of Assets					
10			Q1	Q2	Q3	Q4	
11			%%=Assets.Q1	%%=Assets.Q2	%%=Assets.Q3	%%=Assets.Q4	
12		Quarter Total	\$ -	\$ -	\$ -	\$ -	
13							
14		List of Losses					
15			Q1	Q2	Q3	Q4	
16			%%=Losses.Q1	%%=Losses.Q2	%%=Losses.Q3	%%=Losses.Q4	
17		Quarter Total	\$ -	\$ -	\$ -	\$ -	
18							
19		List of Other Expenses					
20			Q1	Q2	Q3	Q4	
21			%%=Other.Q1	%%=Other.Q2	%%=Other.Q3	%%=Other.Q4	
22		Quarter Total	\$ -	\$ -	\$ -	\$ -	
23							

The bottom of the worksheet has a section for the data that will be imported. Each section has a total row summing all of the values for the section. This formula will expand as ExcelTemplate imports data.

Clipboard		Font			
C12		fx	=SUM(C11:C11)		
	A	B	C	D	E
10			Q1	Q2	Q3
11			%%=Assets.Q1	%%=Assets.Q2	%%=Assets.Q3
12		Quarter Total	\$ -	\$ -	\$ -
13					
14		List of Losses			

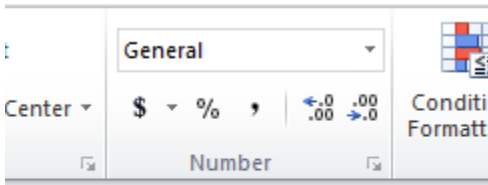
At the top is a grid that calculates the percent in change between each quarter, based on the asset, loss, and other totals. The formulas are set up as "=(Sheet2!B2-Sheet2!A2)/(Sheet2!A2)" for each cell in the grid.

C6 fx =((D22-C22)/(C22))				
	A	B	C	D
4		Assets	#DIV/0!	#DIV/0!
5		Losses	#DIV/0!	#DIV/0!
6		Other	#DIV/0!	#DIV/0!
7				
19	List of Other Expenses			
20			Q1	Q2
21			%%=Other.Q1	%%=Other.Q2
22		Quarter Total	\$ -	\$ -

Number Formats

This section will cover how to add number formatting and the expected behavior for ExcelTemplate as it imports data.

1. Select the 9 cells in the percentage grid at the top of the worksheet (C4:E6).
2. Right click and select "Format Cells..."
3. Select "Percentage" on the Number tab.
4. Click OK
5. Select cells C12:F12, C17:F17, and C22:D22.
6. In the top tool bar, under the 'Number' tab, click the "\$" to apply the currency format.



	G	H	I
--	---	---	---

So far the number formats have been applied to cells that only contain formulas, but number formats also work with cells that contain data markers.

1. Select cells C11:F11, C16:F16, C21:F21.
2. At the top bar, click the "\$" to apply the currency format.

	A	B	C	D	E	F	G
1							
2		Percent Change By Quarter					
3		Type	Q1-Q2	Q2-Q3	Q3-Q4		
4		Assets	#DIV/0!	#DIV/0!	#DIV/0!		
5		Losses	#DIV/0!	#DIV/0!	#DIV/0!		
6		Other	#DIV/0!	#DIV/0!	#DIV/0!		
7							
8		Details					
9		List of Assets					
10			Q1	Q2	Q3	Q4	
11			%%=Assets.Q1	%%=Assets.Q2	%%=Assets.Q3	%%=Assets.Q4	
12		Quarter Total	\$ -	\$ -	\$ -	\$ -	
13							
14		List of Losses					
15			Q1	Q2	Q3	Q4	
16			%%=Losses.Q1	%%=Losses.Q2	%%=Losses.Q3	%%=Losses.Q4	
17		Quarter Total	\$ -	\$ -	\$ -	\$ -	
18							
19		List of Other Expenses					
20			Q1	Q2	Q3	Q4	
21			%%=Other.Q1	%%=Other.Q2	%%=Other.Q3	%%=Other.Q4	
22		Quarter Total	\$ -	\$ -	\$ -	\$ -	
23							

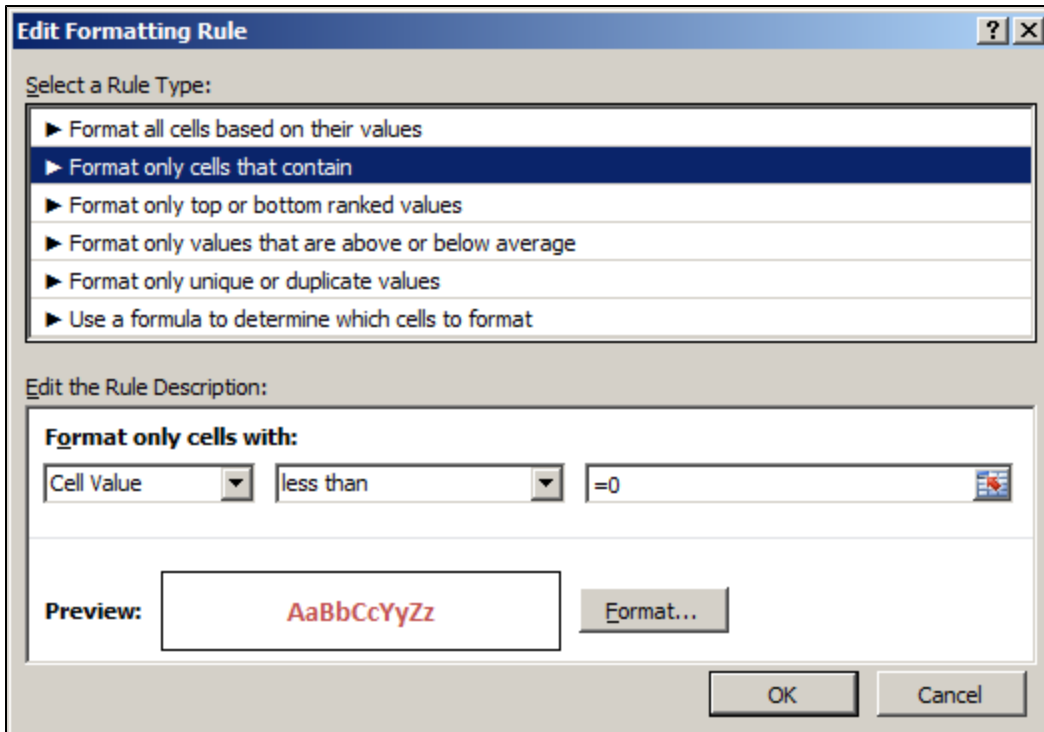
As ExcelTemplate populates the worksheet, the number format will be copied to all the new rows of data that are inserted where the data markers are.

Conditional Formatting

ExcelTemplate will preserve existing conditional formatting. If the conditional format includes a formula, that formula will also be updated as new rows are inserted into the file.

In this case, the conditional formatting will be applied to the 'percent table' and it format negative numbers to be bold and red.

1. On the "Home" tab in Excel, click on "Conditional Formatting"
2. Select "New Rule..."
3. In this tutorial the condition type is "Format only cells that contain..." The rule is "Cell value less than 0"
4. Click on "Format..." Set the text to be dark red. Set the typeface to be bold.
5. Click OK to save the rule.



Adding an ExcelWriter Reference in Visual Studio



Following the Sample Code

In the sample code, the reference to *SoftArtisans.OfficeWriter.ExcelWriter.dll* has already been added to the *CompleteFinancialReport* project.

Create a .NET project and add a reference to the ExcelWriter library.

1. Open Visual Studio and create a .NET project.
 - The sample code uses a web application.
2. Add a reference to *SoftArtisans.OfficeWriter.ExcelWriter.dll*
 - *SoftArtisans.OfficeWriter.ExcelWriter.dll* is located under **Program Files > SoftArtisans > OfficeWriter > dotnet > bin**

Writing the Code

1. Include the *SoftArtisans.OfficeWriter.ExcelWriter* namespace in the code behind

```
using SoftArtisans.OfficeWriter.ExcelWriter;
```

2. In the method that will run the report, instantiate the *ExcelTemplate* object.

```
ExcelTemplate XLT = new ExcelTemplate();
```

3. Open the template file with the *ExcelTemplate.Open* method.

```
XLT.Open(Page.MapPath("../templates//Part2_Financial_Template.xlsx"));
```

4. Create a *DataBindingProperties* object. None of the binding properties will be changed for this tutorial, but *DataBindingProperties* is a required parameter in *ExcelTemplate* data binding methods.

```
DataBindingProperties dataProps = XLT.CreateDataBindingProperties();
```

Data Binding



Following the Sample

Similarly to part 1, we are using CSV that contain values for the data import. Please see [Part 1](#) for more information on using the CSV files and the CSV parser.

1. Get the data for the Assets, Losses, and Other datasets

These calls are to a helper method `GetCSVData` that parses the CSV files and returns a `DataTable` with the values.

```
DataTable dtAssets = GetCSVData("//data//Assets.csv");  
DataTable dtLosses = GetCSVData("//data//Losses.csv");  
DataTable dtOther = GetCSVData("//data//Other.csv");
```

2. Use `ExcelTemplate.BindData` to bind the data for the Assets, Losses, and Other data sets.

```
XLT.BindData(dtAssets, "Assets", bindingProps);  
XLT.BindData(dtLosses, "Losses", bindingProps);  
XLT.BindData(dtOther, "Other", bindingProps);
```

3. Call `ExcelTemplate.Process()` to import all data into the file.

```
XLT.Process();
```

4. Call `ExcelTemplate.Save()` to save the final file.

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

The final output should look something like this:

Percent Change By Quarter				
Type	Q1-Q2	Q2-Q3	Q3-Q4	
Assets	0%	-22%	26%	
Losses	30%	-35%	10%	
Other	-76%	280%	129%	
Details				
List of Assets				
	Q1	Q2	Q3	Q4
	935210	896584	444821	789540
	668779	685210	584579	689547
	856200	845600	886400	825690
	942014	685470	458729	687514
	598700	896522	763351	956874
Quarter Total	\$ 4,000,903.00	\$ 4,009,386.00	\$ 3,137,880.00	\$ 3,949,165.00
List of Losses				
	Q1	Q2	Q3	Q4
	759421	854862	556840	359601
	328569	655654	564332	791417
	665123	951492	568452	382083
	450230	517821	587462	719871
	826974	945821	265411	546564
Quarter Total	\$ 3,030,317.00	\$ 3,925,650.00	\$ 2,542,497.00	\$ 2,799,536.00
List of Other Expenses				
	Q1	Q2	Q3	Q4
	-56943	12037	-66519	-58420
	-65897	-66598	237441	685429
	35789	26874	12574	24564
	368554	95884	75684	-58457
Quarter Total	\$ 281,503.00	\$ 68,197.00	\$ 259,180.00	\$ 593,116.00

Final Code

```
using SoftArtisans.OfficeWriter.ExcelWriter;
...

//Instantiate the template object
ExcelTemplate XLT = new ExcelTemplate();
//Open the file
XLT.Open(Page.MapPath("//templates//Part1_Financial_Template.xlsx"));

//Create data binding properties
DataBindingProperties bindingProps = XLT.CreateDataBindingProperties();

//Get the data from the CSVs. More info about the generic parser is available
//in the project and in the tutorial above.
DataTable dtAssets = GetCSVData("//data//Assets.csv");
DataTable dtLosses = GetCSVData("//data//Losses.csv");
DataTable dtOther = GetCSVData("//data//Other.csv");

//Bind each datatable
XLT.BindData(dtAssets, "Assets", bindingProps);
XLT.BindData(dtLosses, "Losses", bindingProps);
XLT.BindData(dtOther, "Other", bindingProps);

//Call process to import data to file
XLT.Process();

//Call save
XLT.Save(Page.Response, "temp.xlsx", false);
```

Downloads

You can download the code for the Financial Report here.

- [ExcelWriter_Basic_Tutorials.zip](#)

Next Steps

[Continue to Part 3: Combine Reports with CopySheet](#)