Using Grouping and Nesting with ExcelTemplate

Starting in version 7.1 of ExcelWriter, you can use [grouping and nesting markers|Creating Data Markers#grouping] to display flat data in a grouped and nested format using the ExcelTemplate object.

Introduction

In order to help organize hierarchical data in a more easily readable format, ExcelTemplate now features grouping and nesting options. The basic feature of grouping and nesting is that ExcelTemplate can remove unnecessary repeated values from a field if it is given hierarchical data in a flat format. This basic functionality is explained in further detail below, while advanced features such as headers and footers are discussed on the Advanced Grouping and Nesting page.

Creating a Data Set for grouping

Though grouping and nesting is meant to show hierarchical data, the input data for a template with grouping and nesting must be in flat format. This means that a data set for nesting and grouping will have repeated values for fields that are higher in the hierarchy. For this tutorial, the data will look like this:

Territory Name	Salesperson	Product Category	Product Subcategory	Color	Unit Price	Order Qty	Line Total
Central	Jillian Carson	Bikes	Mountain Bikes	Black	$2024.99	1	$2024.99
Central	Jillian Carson	Bikes	Mountain Bikes	Black	$2024.99	1	$2024.99
Central	Jillian Carson	Bikes	Mountain Bikes	Silver	$2039.99	1	$2039.99
Central	Jillian Carson	Clothing	Socks	White	$5.70	3	$17.10
Northwest	David Campbell	Accessories	Helmets	Red	$20.19	2	$40.37
Northwest	David Campbell	Accessories	Helmets	Blue	$20.19	1	$20.19
Northwest	David Campbell	Bikes	Mountain Bikes	Black	$2024.99	2	$4049.99
Northwest	David Campbell	Bikes	Mountain Bikes	Black	$2024.99	1	$2024.99

Grouping Code

Grouping code is no different from regular ExcelTemplate code. In this tutorial, we use the following ExcelTemplate code:

```
DataSet ds = GetData(); //--- Get the Sales data
ExcelTemplate xlt = new ExcelTemplate(); //--- Create a new ExcelTemplate object
xlt.Open(Page.MapPath(@"templates\GroupingTemplate.xlsx")); //--- Open the template

//--- Create the DataBindingProperties object
DataBindingProperties props = xlt.CreateDataBindingProperties();

//--- Bind the data to the template
xlt.BindData(ds, "AdventureWorks", props);

//--- Process the template to import the data
xlt.Process();

//--- Stream the output back to the user
xlt.Save(Page.Response, "Grouped Sales Report.xlsx", false);
{code}
{code:vb.net|title=VB.NET}
Dim ds As DataSet = GetData() '--- Get the Sales data
Dim xlt As New ExcelTemplate() '--- Create a new ExcelTemplate object
xlt.Open(Page.MapPath("templates\GroupingTemplate.xlsx")) '--- Open the template

'--- Create the DataBindingProperties object
Dim props As DataBindingProperties = xlt.CreateDataBindingProperties()

'--- Bind the data to the template
xlt.BindData(ds, "AdventureWorks", props)

'--- Process the template to import the data
xlt.Process()

'--- Stream the output back to the user
xlt.Save(Page.Response, "Grouped Sales Report.xlsx", False)
```

Grouping by a single group

The first element needed for a grouping template is a data row that contains all of the data markers for your data set, as shown below:

If you bound the data above to this template, it would display exactly as you see it above -- in flat form. You can group based on the first column -- Territory Name -- by adding a *%%group* tag to the cell above the data marker for that column. If you add any *%%group* tag, you must also add an *%%endgroup* tag below the data markers in order to mark the end of the grouping section:

	A	B	C	D	E	F	G
1	PostOne						
2							
3							
4	**Territory Sales Report**						
5	**Territory Name**	**Salesperson**	**Product Category**	**Product Subcategory**	**Color**	**Unit Price**	**Order Qty**
6	%%group						
7	%%=[AdventureWc	%%=[AdventureV	%%=[AdventureW	%%=[AdventureWor	%%=[Adventure	%%=[Adventu	%%=[Adver
8	%%endgroup						
9	**Grand Total**						

The resulting spreadsheet should look like the one below, with all of the extra repeated values from the Territory Name field removed:

	A	B	C	D	E	F	G
1	PostOne						
2							
3							
4	**Territory Sales Report**						
5	**Territory Name**	**Salesperson**	**Product Category**	**Product Subcategory**	**Color**	**Unit Price**	**Order Qty**
6	Canada	José Saraiva	Accessories	Helmets	Black	$20.19	
7		José Saraiva	Accessories	Helmets	Blue	$20.19	
8		José Saraiva	Accessories	Helmets	Blue	$20.19	
9		José Saraiva	Accessories	Helmets	Red	$20.19	
10		José Saraiva	Bikes	Mountain Bikes	Black	$2,024.99	
11		José Saraiva	Bikes	Mountain Bikes	Black	$2,024.99	
12		José Saraiva	Bikes	Mountain Bikes	Black	$2,024.99	
13		José Saraiva	Bikes	Mountain Bikes	Black	$2,024.99	
14		José Saraiva	Bikes	Mountain Bikes	Silver	$2,039.99	
15		José Saraiva	Bikes	Road Bikes	Red	$2,146.96	
16		José Saraiva	Bikes	Road Bikes	Red	$2,146.96	
17		José Saraiva	Bikes	Road Bikes	Red	$2,146.96	

Notice that the rows with the *%%group* and *%%endgroup* tags are removed as well. Be sure not to place any content in these rows, as it will be automatically removed when data is bound to the template.

Nested grouping with multiple groups

You can group by multiple groups by placing extra **%%group*** markers in more columns. Additional **%%endgroup*** markers are unnecessary. Each of the new fields will be grouped in order of hierarchy, from left to right unless otherwise specified. For example, if you want to group first by Territory Name, then by Salesperson, place an additional **%%group*** marker in the second column next to your first **%%group*** marker:

	A	B	C	D	E	F	G
		PostOne					
1							
2							
3							
4	**Territory Sales Report**						
5	Territory Name	Salesperson	Product Category	Product Subcategory	Color	Unit Price	Order Qty
6	%%group	%%group					
7	%%=[AdventureWo	%%=[AdventureV	%%=[AdventureW	%%=[AdventureWor	%%=[Adventure	%%=[Adventu	%%=[Adver
8	%%endgroup						
9	**Grand Total**						

The resulting spreadsheet will be grouped first by Territory Name, then by Salesperson:

	A	B	C	D	E	F	G
		PostOne					
1							
2							
3							
4	**Territory Sales Report**						
5	Territory Name	Salesperson	Product Category	Product Subcategory	Color	Unit Price	Order Qty
48			Components	Mountain Frames	Black	$714.70	
49			Components	Mountain Frames	Black	$714.70	
50			Components	Mountain Frames	Black	$809.76	
51			Components	Mountain Frames	Silver	$722.59	
52			Components	Mountain Frames	Silver	$818.70	
53			Components	Road Frames	Black	$178.58	
54	Central	Jillian Carson	Bikes	Mountain Bikes	Black	2024.994	
55			Bikes	Mountain Bikes	Black	2024.994	
56			Bikes	Mountain Bikes	Silver	2039.994	
57			Clothing	Socks	White	5.7	
58	Northwest	David Campbell	Accessories	Helmets	Blue	20.1865	
59			Accessories	Helmets	Red	20.1865	
60			Bikes	Mountain Bikes	Black	2024.994	
61			Bikes	Mountain Bikes	Black	2024.994	
62			Bikes	Mountain Bikes	Black	2024.994	

You can group even further by adding additional **%%group*** tags:

	Territory Name	Salesperson	Product Category	Product Subcategory	Color	Unit Price	Order Qty
	Territory Sales Report						
6	%%group	%%group	%%group	%%group			
7	%%=[AdventureWo	%%=[AdventureV	%%=[AdventureW	%%=[AdventureWor	%%=[Adventure	%%=[Adventu	%%=[Adven
8	%%endgroup						
9	**Grand Total**						

The resulting spreadsheet:

	Territory Name	Salesperson	Product Category	Product Subcategory	Color	Unit Price	Order Qty
	Territory Sales Report						
6	Canada	José Saraiva	Accessories	Helmets	Black	20.1865	
7					Blue	$20.19	
8					Blue	$20.19	
9					Red	$20.19	
10			Bikes	Mountain Bikes	Black	$2,024.99	
11					Black	$2,024.99	
12					Black	$2,024.99	
13					Black	$2,024.99	
14					Silver	$2,039.99	
15				Road Bikes	Red	$2,146.96	
16					Red	$2,146.96	
17					Red	$2,146.96	
18					Red	$874.79	
19					Red	$874.79	
20					Red	$874.79	
21					Red	$874.79	

Additional Functionality

Additional functionality such as headers and footers, are explained in the Advanced Grouping and Nesting page.