



Flexcel Reports Designers Guide

Documentation: May, 2010
Copyright © 2010 by tmssoftware.com bvba
Web: <http://www.tmssoftware.com>
Email : info@tmssoftware.com

Table of contents

- Introduction..... 3
- Report Elements 4
 - Tags 4
 - ##dataset##field 4
 - #.variable#. 4
 - “...delete row...” and “...page break...” 4
 - Named ranges 5
- Range notes 8
- Master detail..... 9
- Deleting Rows 10
- Fixed Bands 12
- Appendix: Using FlexCel designer 13

Introduction

This document has two different target audiences, developers using FlexCel and final power users that want to customize their reports. If you are a developer, please look at the developers guide before this one.

Report Elements

There are two concepts you should understand to create or modify a report:

Tags

Tags are special symbols you can write in a template that will be replaced by values from your application when generating the report. You can write the following tags in a template:

```
##dataset##field
```

This tag will be replaced by a value in a TDataSet (or TFlxMemTable) component named “Dataset” in the datamodule where the TFlexCelReport component is.

If you just write it into an xls sheet, the value of the current record will be used. If you drop it inside a “__dataset__” named range, all the dataset will be dumped to the template. See “Named Ranges” section below for more details.

```
#.variable#.
```

This tag will be replaced by a custom value you define in your application. There are two ways to give a value to “Variable” so it gets replaced:

1. You can use a line of code like:

```
FlexCelReport.Values['variable'] := 'hello';
```

Before calling

```
FlexCelReport.Run;
```

This is normally the simplest way to define a report variable.

2. You can create a published function or published property named “Variable” in the datamodule where the TFlexCelReport component is, that will return the value of the variable when asked. While this option needs more work than the first, it also allows you more flexibility. Values set with the first method are fixed through all the report generation, while here you can calculate a different value each time FlexCel needs to replace the variable. This allows you to create user defined functions, for example, you can define a method “TotalCost” that will return the fixedcost plus variablecost fields in a database. You can also pass variables to these published methods.

Note that #.variables#. can be simple variables, or they might refer also to an array of variants. If they are an array, then you must write in the template something like “#.variable#.index1#.index2...” where “indexn” is the n index to the array. Also when calling published methods you can include multiple values the same way.

Take a look at the “Getting Started Reports”, “Array of Variants” and “User defined functions” demos for more information.

```
“...delete row...” and “...page break...”
```

Those tags should always be placed in column “A” in the report, and they will delete a row or add a page break at the current position. See “Deleting Rows” section below for more information on how to use the “Delete row” tag, and make sure **you are writing three real dots “...” and not an ellipsis “...”**. Look at the note in “Deleting rows” section below.



Sometimes you might want to write more than a tag in a single cell. Say for example that you have the First name and Last name of a customer in two different columns in a table, and you want to print them together in one cell. In order to use more than one tag you should surround them by “<” and “>” signs.

For example, you could write “First Name:<##customer##firstname>, Last Name: <##customer##lastname>” in a cell, and it will be replaced by the corresponding values. You can even format the tags different, for example write “<##customer##firstname>” in bold blue text, and it will be bold blue, while last name will be normal.

Named ranges

While Tags allow you to replace expressions inside a sheet, with them alone we can only make “Fill in the blanks” type reports. That is, reports that are static, like a form, and where cells with tags will be replaced with their corresponding values.

Now we are going to introduce the concept of “Band” A Band is just a range of cells that is repeated for each record of a table. Imagine that you define a Band on the cell range A1:C1, and associate it with the table “Customer”. Then, on cells A1:C1 you will have the first customer, on cells A2:C2 the second and so on. All cells that were previously on A2:C2 will be moved down after the last record of the dataset.

If table customer has six registers, you will get something like:

Template:

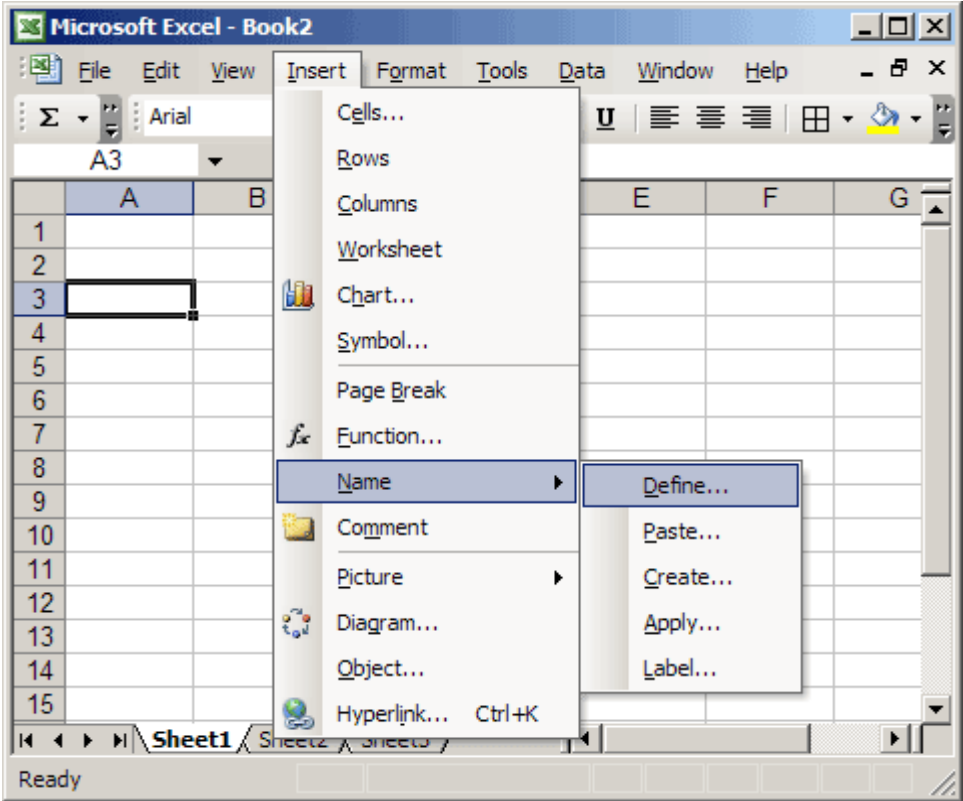
	A	B	C	D
1	<#Customer.CustomerName>	<#Customer.Id>	<#Customer.Age>	
2	This is some text below the band.			
3				
4				

Generated Report:

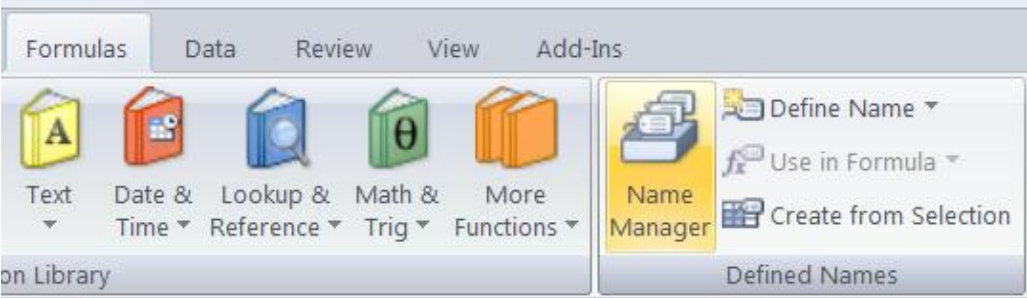
	A	B	C	D
1	Customer1	1	32	
2	Customer2	2	33	
3	Customer3	3	12	
4	Customer4	4	34	
5	Customer5	5	44	
6	Customer6	6	98	
7	This is some text below the band.			
8				
9				

On FlexCel we use Named Ranges to indicate the bands. If you are not used to Excel Named Ranges, take some time to familiarize with them, as they are one of the things that confuses most people starting with FlexCel. Different from tags, that you can immediately see when you open a workbook, named ranges are a little more hidden.

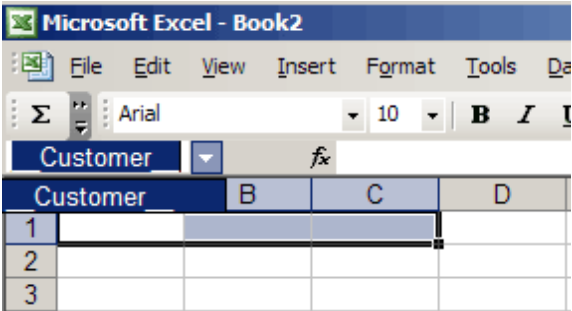
To create a Band on A1:C1, we would go to Menu->Insert->Name->Define:



Or In Excel 2007, go to “Formulas” tab in the ribbon, and press “Name Manager”:



Once there, we can define a Band __Customer__ on cells A1:C1. And once the name is defined, we can easily see it on the Names combo:



Note the “__” at the beginning and at the end of the range name. We use this to indicate FlexCel that this is an horizontal range that inserts full rows down.

The rest of the name (“Customer”) should be the name of an existing dataset in your datamodule.

Range notes

Note 1: Don't intersect ranges, or you are going to get strange results. Each range should be either completely inside or completely outside the others.

Note 2: The ranges affect the entire row that will be copied. Never put one range to the right or left of the other.

Note 3: It is possible to define some __XX__ ranges that don't correspond to any dataset. These ranges are not going to be copied, but values will be substituted

Note 4: Always define some Range that includes all the others, so the component knows where to make the replacements. FlexCelReport always will look for the biggest range to begin replacing values. We usually call this range __MAIN__, but you can call it as you want, as long as it begins with “__” and ends with “__”

Master detail

Named ranges can be placed inside others, and a master detail relationship will be automatically created. For example, If you define a range “__Customer__” and inside it a range “__Orders__” and there is a relationship created on the application between “Customer” and “Order” tables, it will automatically group your orders by customer.

On the following example, the yellow cells are the range “__Customer__” and the blue ones are the range “__Orders__”

	A	B	C	D	E	F
1						
2	Customer:	<#Customer.Name>				
3		Orders				
4		<Order.Id>				
5						
6						

After running this report, you will get something similar to:

	A	B	C	D	E	F
1						
2	Customer:	Customer1				
3		Orders				
4		Order1 for Customer1				
5		Order2 for Customer1				
6		Order3 for Customer1				
7	Customer:	Customer2				
8		Orders				
9		Order1 for Customer2				
10		Order2 for Customer2				
11	Customer:	Customer3				
12		Orders				
13		Order1 for Customer3				
14		Order2 for Customer3				
15		Order3 for Customer3				
16						
17						

As you can see, Orders are filtered for each customer, based on the Data Relationship defined on the application, and on the nesting on the ranges. In general, any range that is inside another is filtered by all of its parents. You can have as many levels of master-detail relationships as you wish, and each master band filters all of its children.

For example, if we wanted to group the customers by country we could define a __Countries__ named range on A1:F6, and it would automatically filter the data on its child and grand-child. (Customer and Order)

There is a special Table that if present filters all the others on the sheet, acting as a parent of all the named ranges on the sheet. This is the table that you define on the name of the sheet, when doing a multiple sheet report. You can see the multiple sheet report demo for more information, there every table on each sheet is filtered by category.

Deleting Rows

One issue that might appear when defining named ranges is how formulas on other ranges change when inserting the new cells.

Let's imagine we want to make a simple report on a list of items and their prices.

So, we create a new template, define a title row, and insert a named range on A2:B2 to propagate the data. But we also want to know all the total cost of all items, so we add a formula on B3:

	A	B	C
1	Item	Price	
2	<#Item.Name>	<#Item.Price>	
3		=Sum(B2:B2)	
4			

When you run this report, rows will be inserted between row 2 and 3, but the formula Sum(B2:B2) won't change. Nothing has been inserted between B2 and B2, so the sum range will remain constant.

So, we need to have a Sum range that can expand. We will define:

	A	B	SUM(number1; [number2]; ...)		E
1	Item	Price			
2	<#Item.Name>	<#Item.Price>			
3					
4		=SUM(B2:B3)			
5					

Now, when rows are inserted between row 2 and 3, the formula will be updated to reflect the new rows.

	A	B	SUM(number1; [number2]; ...)		E
1	Item	Price			
2	Apple	20			
3	Banana	10			
4	Orange	5			
5					
6		=SUM(B2:B5)			
7					

On this particular case, this solution might be enough. Just leave an empty row after the range so formulas expand, and then you can hide the extra row or just leave it there.

But, if we were for example creating a chart, this extra row will be on the chart too. There is a simple solution for this, and that is to write "...delete row..." in cell A6.

All the rows that have in the "A" column the text "...delete row..." (without any leading or trailing spaces and all in lowercase) will be deleted after the report is run. An use of this is if you want to create a Pivot Table or a Chart, and you can see it in the PivotDemo.xls template.



Important: With the standard configuration, Excel XP automatically replaces the 3 dots “...” with an ellipsis char “...”. But even when they look the same they are not the same. If you are having problems with FlexCel understanding “delete rows”, check that you have real dots in there.


Fixed Bands

By default, FlexCel will always insert cells when expanding ranges, and this is what you would normally want. If you have a template:

```
A1:Title
A2:##dataset.field
A3:Footer
```

You would expect that the generated report will have the Footer for example on cell A33 (if we had 30 data records), but not on A3.

But there is a situation where this is not what you expect, and this is on Fixed Form reports. Let's imagine that you want to fill out a form with FlexCel. Most fields will be just simple expressions, not related to datasets, but we might have a table "TopOrders" too:

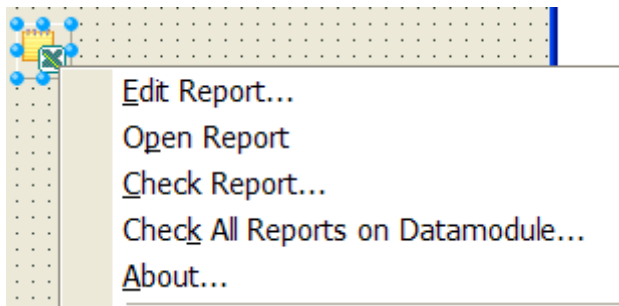
	A	B	C	D	E	F	G	H
1								
2			<##Employee##FirstName> <##Employee##LastName> Data <i>An Example on how to use bands inside fixed forms</i>					
3								
4								
5								
6								
7	Name:	<##Employee##FirstName> <##Employee##LastName>						
8	Title:	<##Employee##Title>						
9								
10	Top 10 orders from Employee							
11	Order	Customer	Shipped Date			Freight		
12	1	<##TopOrders##ShipName>	<##TopOrders##ShippedDate>			<##TopOrders##Freight>		
13	2							
14	3							
15	4							
16	5							
17	6							
18	7							
19	8							
20	9							
21	10							
22								
23	This is a Fixed form report and this line is expected to always print on this particular place					Total	0,00	
24								
25	Note that we removed Autofit from rows so they do not change with the cell contents.							
26								
27	Sign:							
28	<div style="border: 1px solid black; width: 200px; height: 40px; margin: 0 auto;"></div>							
29								
30								
31								
32								

Here, no matter if TopOrders has 1 record, two or 10 (it should not have more than 10) you want the "Total" line to be at row 23. You cannot do this with normal ranges, since you would be inserting rows. For this you can set the property "ReportModeRows" in the FlexCelReport component to "frm_DontCopyAndDontInsert".

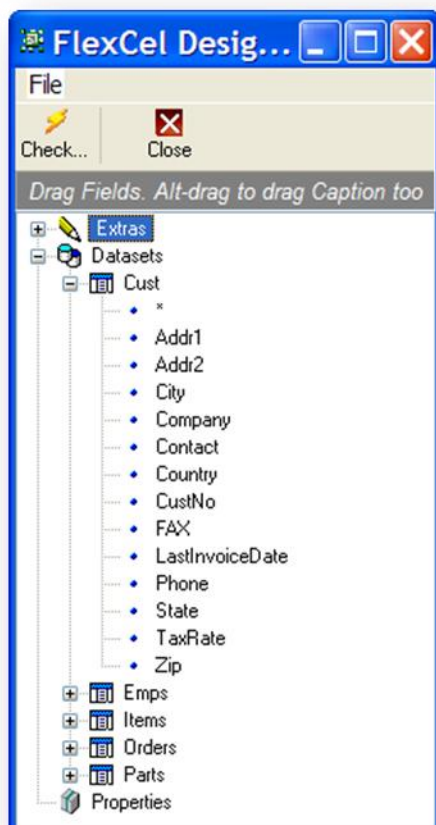
Appendix: Using FlexCel designer

While you can create your reports only with Excel, writing all the tags can be a time consuming task. FlexCel designer is a little tool created to help on this.

You can invoke the FlexCel designer by double-clicking in a TFlexCelReport component, or by right clicking on it and choosing "Edit Report":



When you launch FlexCel designer, a screen similar to the following will open:



On the listbox you can select a field, a tag, a report variable or extra identifiers as "...delete row...", and drag and drop them to Excel or Open Office.

If you press "Alt" before beginning to drag a field, the field and the field name