

《Microsoft SQL Server 2008 Reporting Services 应用系列》

本教程说明及版权申明

- ❖ 《Microsoft SQL Server 2008 Reporting Services 应用系列》是为了方便广大 Reporting Services 爱好者共同学习，特投入大量心血及时间撰写的系列课程，并免费提供下载阅读。所有相关文档版权均属本人所有。
- ❖ 本教程是由本人结合官方文档及实际开发经验整理出的一部原创作品。为了方便大家及时学习，会加快撰写速度并定期发布，时间仓促，难免有误，敬请谅解～～
- ❖ 本文档受中华人民共和国版权法保护，任何未经同意的复制和抄袭行为将追究法律责任，任何其他个人或组织均不得以任何形式将本文档用作商业用途，违者必究。
- ❖ 如果对本文档有任何疑问及建议，请在官方博客 <http://www.cnblogs.com/stevenshi/> 留言或发邮件到 sqlserver2008.ms@gmail.com 。

Microsoft SQL Server 2008 Reporting Services 应用系列（三）开发篇

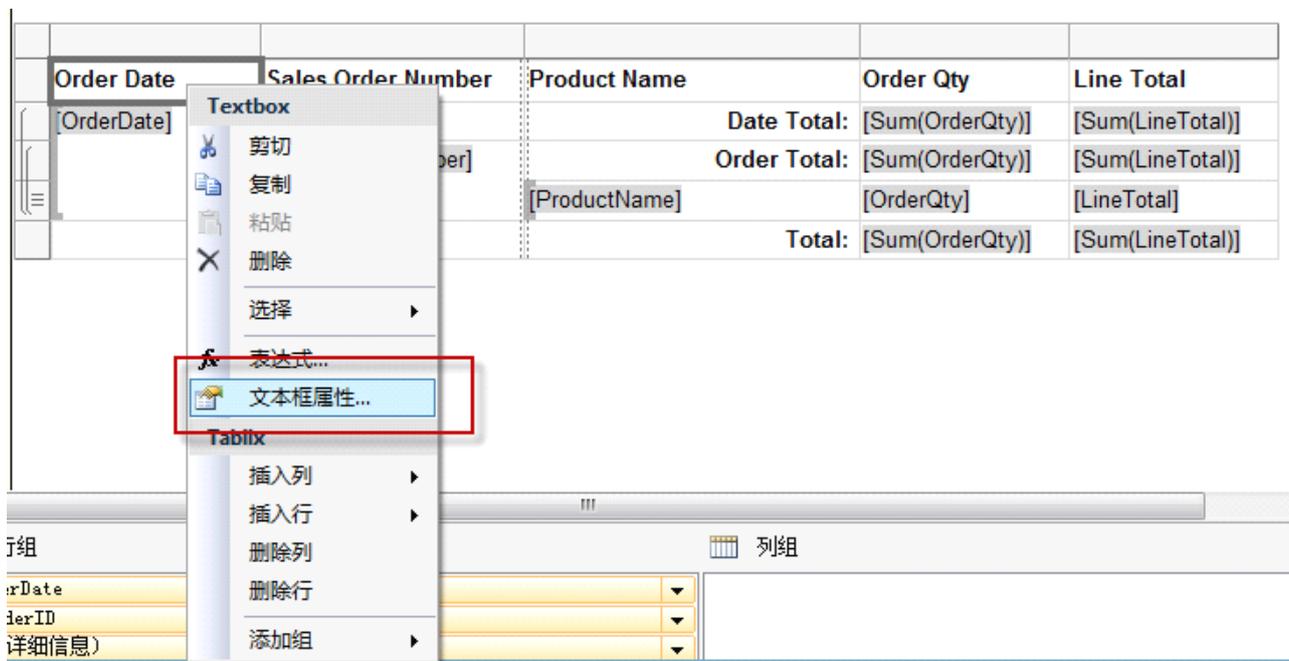
“表” 控件内容补充

上一篇讲了 Tablix 控件的使用，这一篇将继续讲解控件的使用，在上一篇我们遗漏了一些 Tablix 的功能没有讲到，在此先作一下补充：大家还记得在上篇中我们的实例（一）的效果吗？

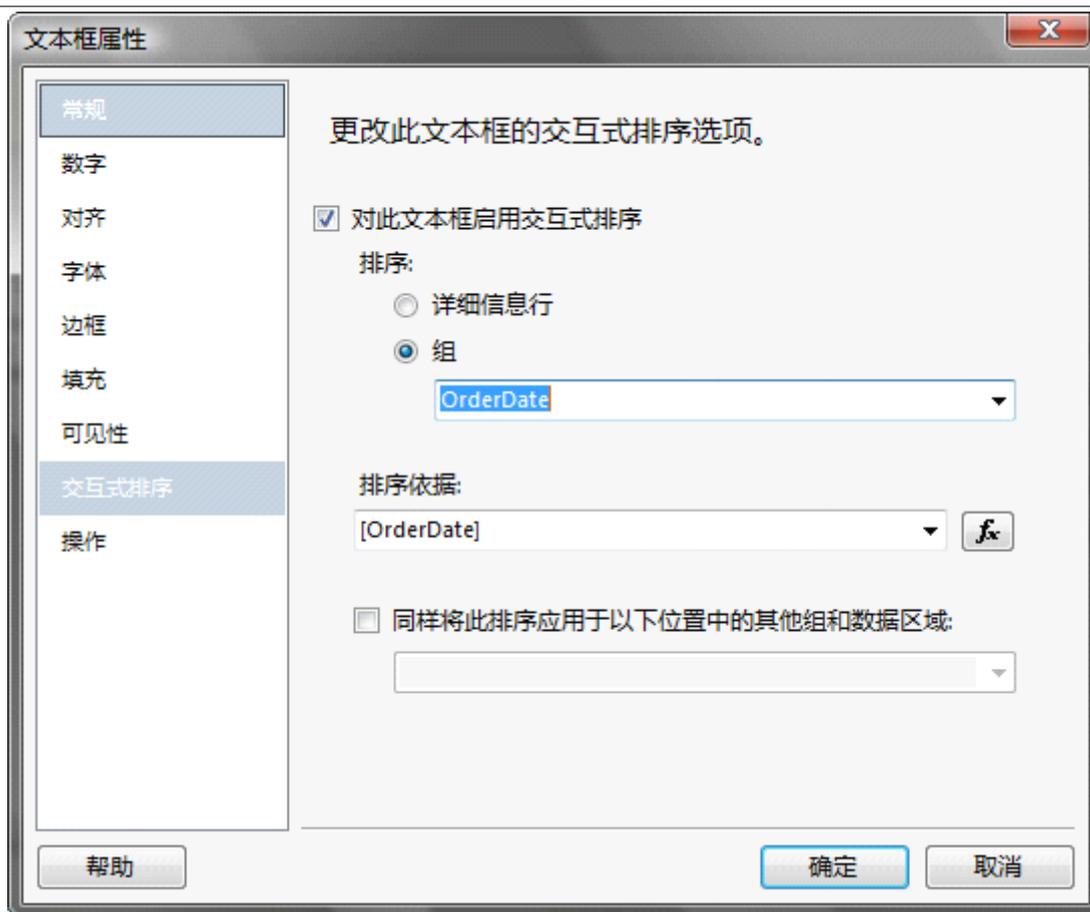
Order Date	Sales Order Number	Product Name	Order Qty	Line Total
2002/2/2 0:00:00			Date Total: 6	18590.448200
2002/2/3 0:00:00			Date Total: 6	21469.620000
2002/2/4 0:00:00			Date Total: 7	16410.374600
2002/2/5 0:00:00			Date Total: 3	10734.810000
2002/2/6 0:00:00			Date Total: 9	20687.742800
			Total: 31	87892.995600

有的朋友提出，Order Date 默认是由小到大排序的，我们如何让用户来控制呢？

OK，让我们来学习一下，其实这很简单，也是我们在上篇中没有介绍的功能。首先我们还是来看一下如何设置：



右键 Order Date 的列标题，在菜单中选择“文本框属性”。进入设置界面：



我们选择“交互式排序”选项卡，然后勾选“对此文本框启用交互式排序”，选择我们需要排序的组，再选择排序依据。如果想把该规则应用到其他地方，可以勾选“同样将此排序应用于以下位置中的其他组和数据区域”。我们点击“确定”来看一下效果：

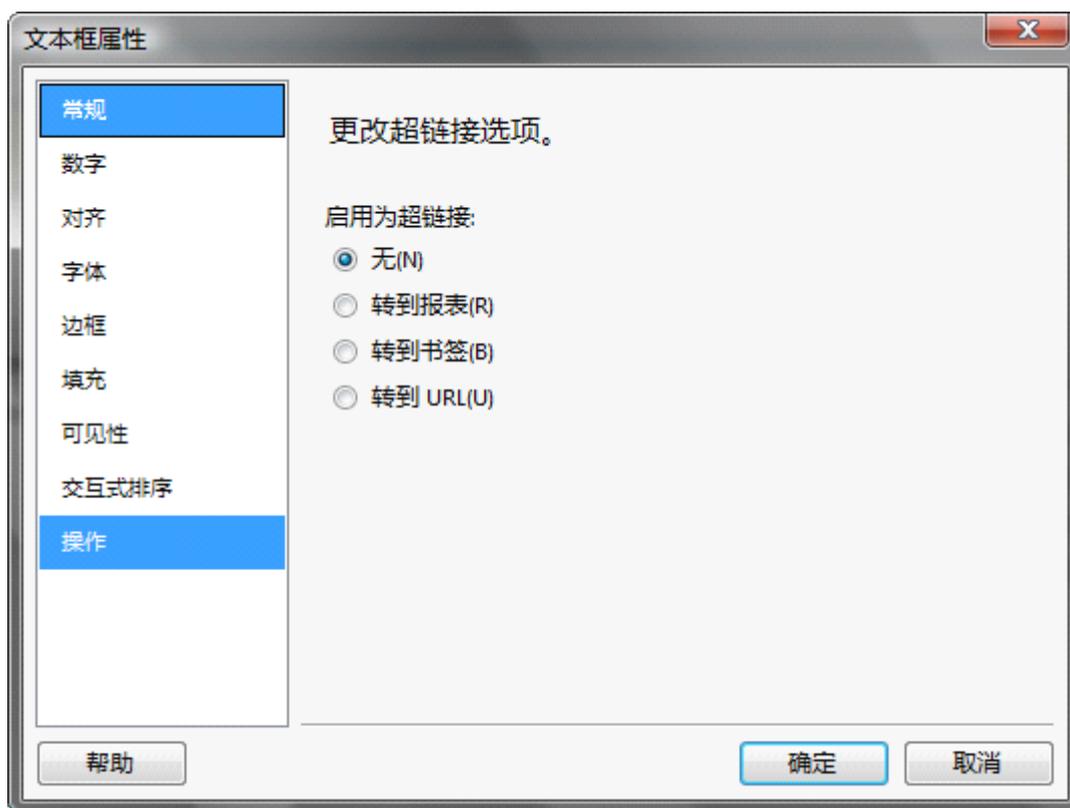
Order Date	Sales Order Number	Product Name	Order Qty	Line Total
2002/2/2 0:00:00			Date Total: 6	18590.448200
2002/2/3 0:00:00			Date Total: 6	21469.620000
2002/2/4 0:00:00			Date Total: 7	16410.374600
2002/2/5 0:00:00			Date Total: 3	10734.810000
2002/2/6 0:00:00			Date Total: 9	20687.742800
			Total: 31	87892.995600

大家可以看到在 Order Date 多出了排序按钮。我们点击试试：

Order Date	Sales Order Number	Product Name	Order Qty	Line Total
2002/2/6 0:00:00			Date Total: 9	20687.742800
2002/2/5 0:00:00			Date Total: 3	10734.810000
2002/2/4 0:00:00			Date Total: 7	16410.374600
2002/2/3 0:00:00			Date Total: 6	21469.620000
2002/2/2 0:00:00			Date Total: 6	18590.448200
			Total: 31	87892.995600

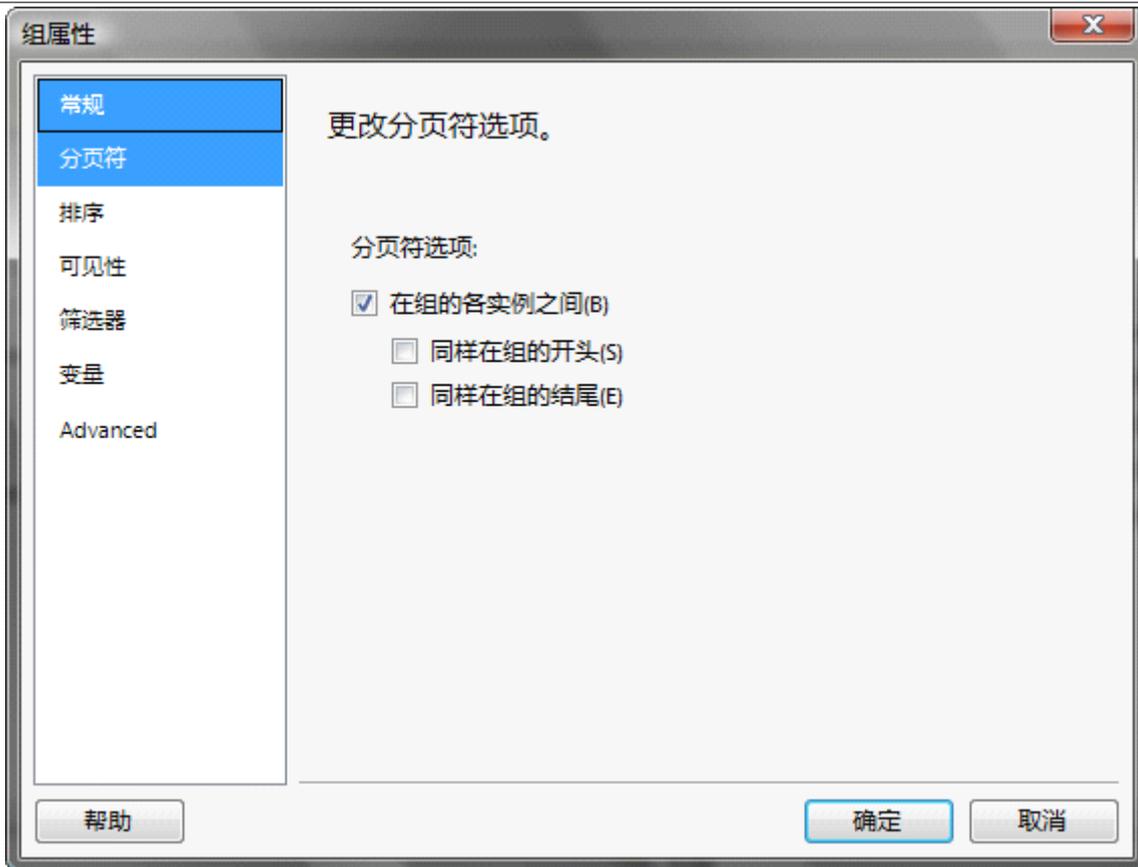
可以看到订单时间已经从小到大，变成了从大到小。当然你也可以把排序功能应用于其他的列，或分组，详细数据等。

我们再介绍一下“文本框属性”窗口的“操作”选项卡。如图

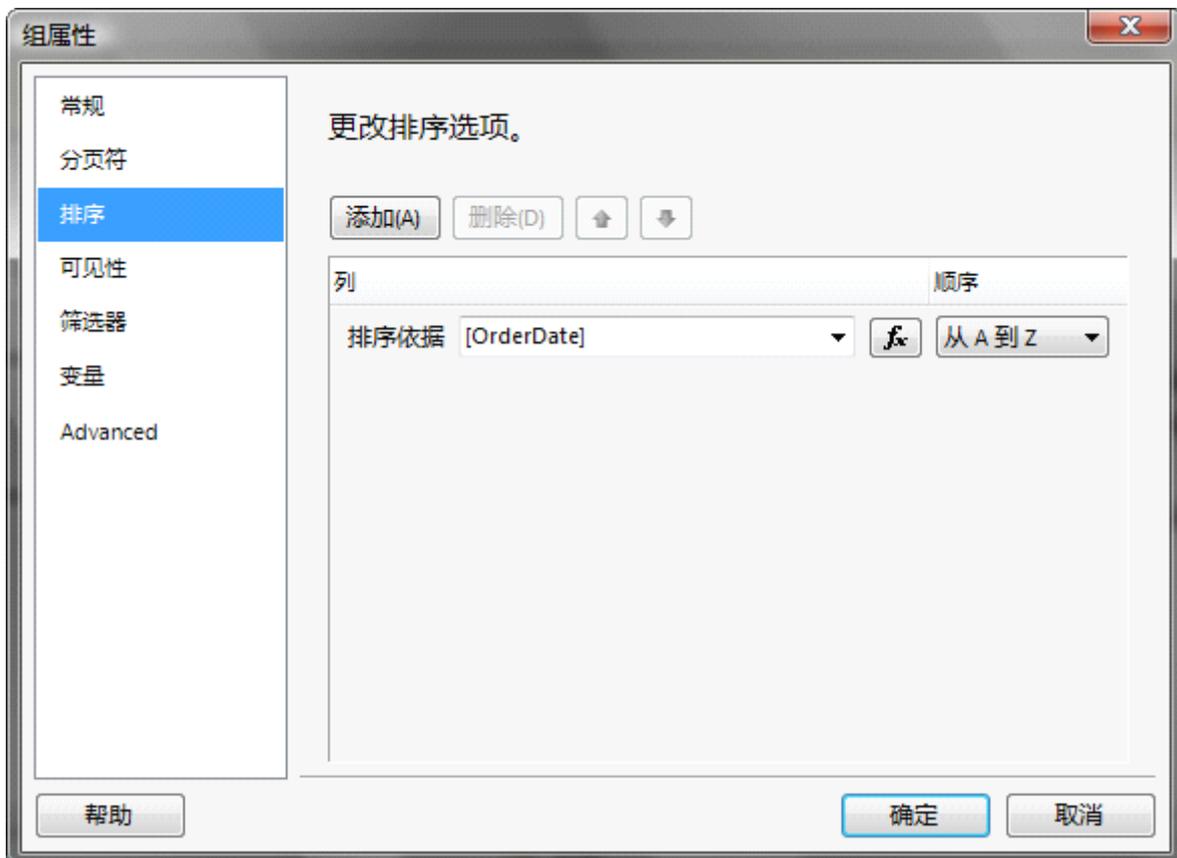


在此可以为该文本框启用超链接功能，可以转到报表，书签，URL。应用场景比如在订单 ID 上加入超链接，点击跳转到明细报表。我们在后面的章节中会用到这些功能，大家可以先熟悉一下。至于其他的选项卡（数字，对齐，字体，边框，填充），大家可以自行熟悉。

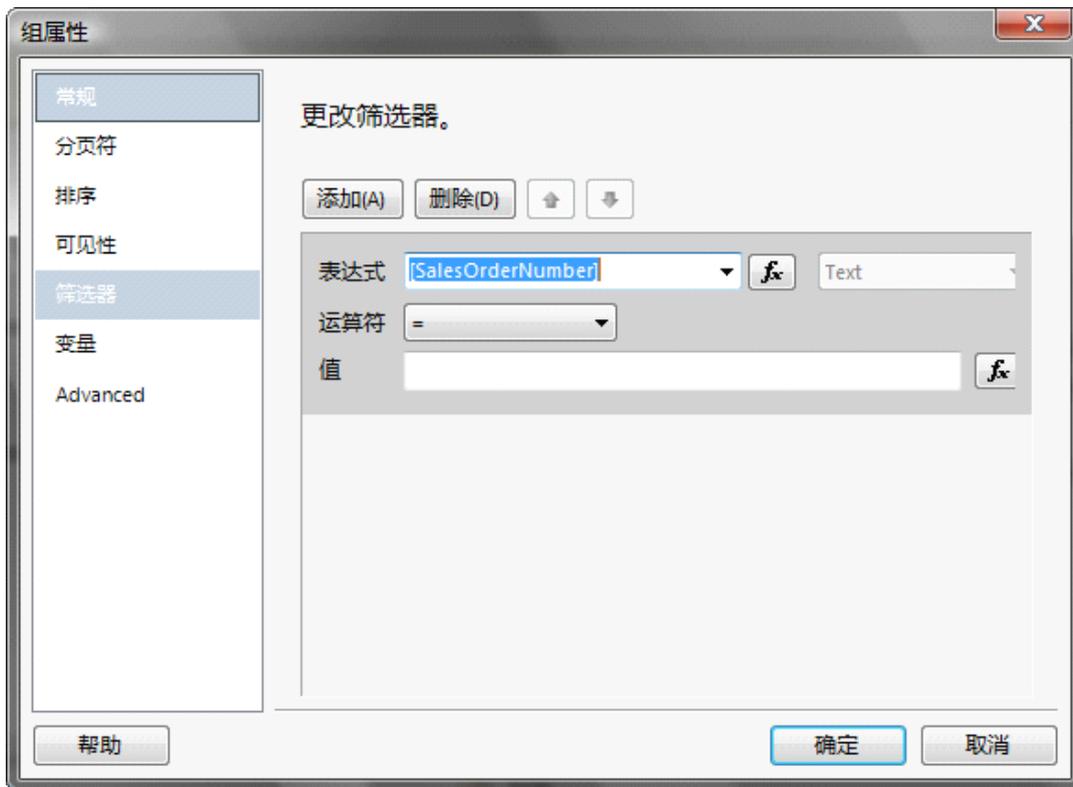
下面再介绍一下组属性：



“分页符”选项卡是做什么的呢？明知故问～～，肯定是分页了，你可以设置分页符放在组的开头还是结尾。

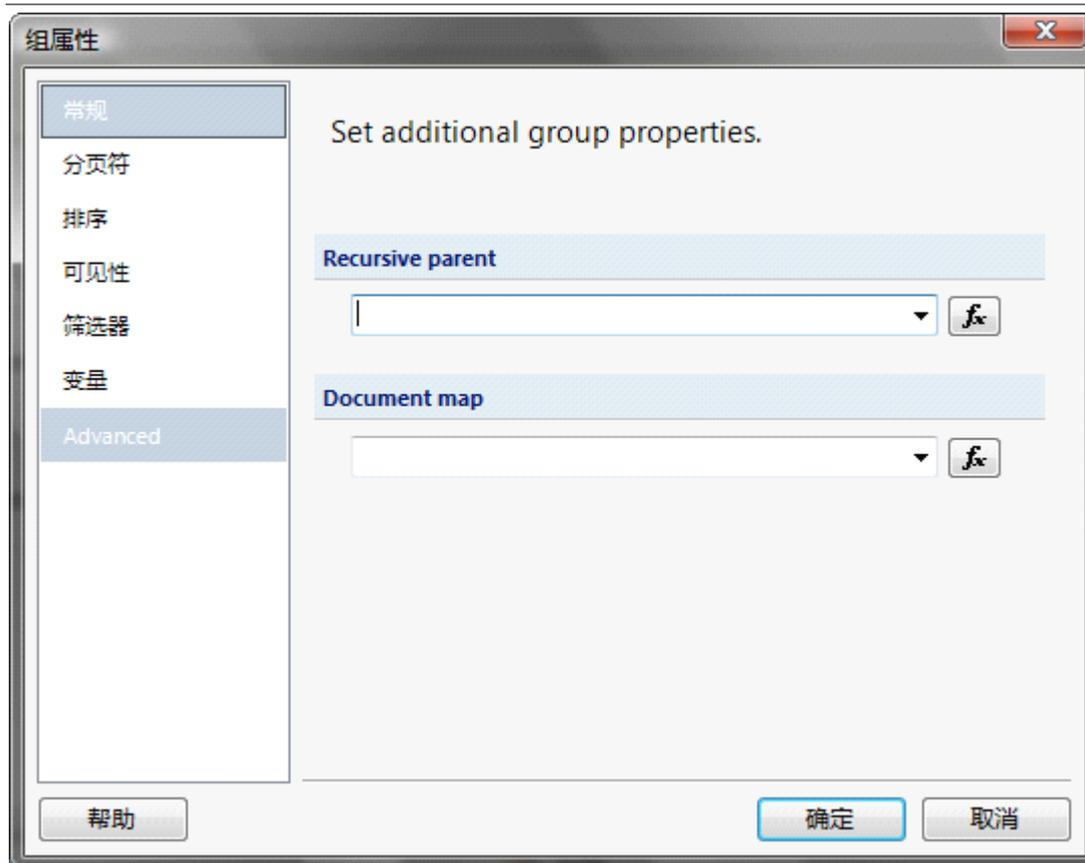


组的排序依据默认和分组依据相同，当然你可以在此自己定义规则。可见性我们在上一篇已经介绍过，再看看筛选器：



若要细化组中的数据，可以创建筛选器表达式。报表将在应用报表数据集、数据区域和父组的筛选器之后，才应用组筛选器。“变量”选项卡可以在此定义一些变量，供子组使用。

“Advanced”选项卡可以创建递归结构的组结构。



若要在 Tablix 数据区域中创建一个递归层次结构，必须将组表达式设置为指定子数据的字段，并将该组的 Parent 属性设置为指定父数据的字段。数据集的数据要符合递归的要求，不要出现死循环。

对 Taxlix 的补充就到这里，如果还有其他问题，欢迎给我留言或 Email 联系我。

“矩阵”的使用

下面让我们继续学习新知识~~我们先学习矩阵的使用。上边有了对“表”控件的理解，矩阵的使用学起来将会很轻松~~

下面我们先来学习一下矩阵的基本功能。首先新建报表，新建数据集：SQL 脚本如下：

```
SELECT year(OrderDate) as Oyear,PC.Name AS ProductCategoryName,
PSC.Name as ProductSubcategoryName,CountryRegionCode,
ST.[Group] AS SalesTerritoryGroup,LineTotal
FROM Sales.SalesOrderHeader O
INNER JOIN Sales.SalesOrderDetail OD
ON O.SalesOrderID=OD.SalesOrderID
INNER JOIN Production.Product P
ON OD.ProductID=P.ProductID
INNER JOIN Production.ProductSubcategory PSC
ON P.ProductSubcategoryID=PSC.ProductSubcategoryID
INNER JOIN Production.ProductCategory PC
ON PSC.ProductCategoryID=PC.ProductCategoryID
```

```
INNER JOIN Sales.SalesTerritory ST
```

```
ON O.TerritoryID=ST.TerritoryID
```

```
where OrderDate>'20031001' AND OrderDate<'20041001'
```

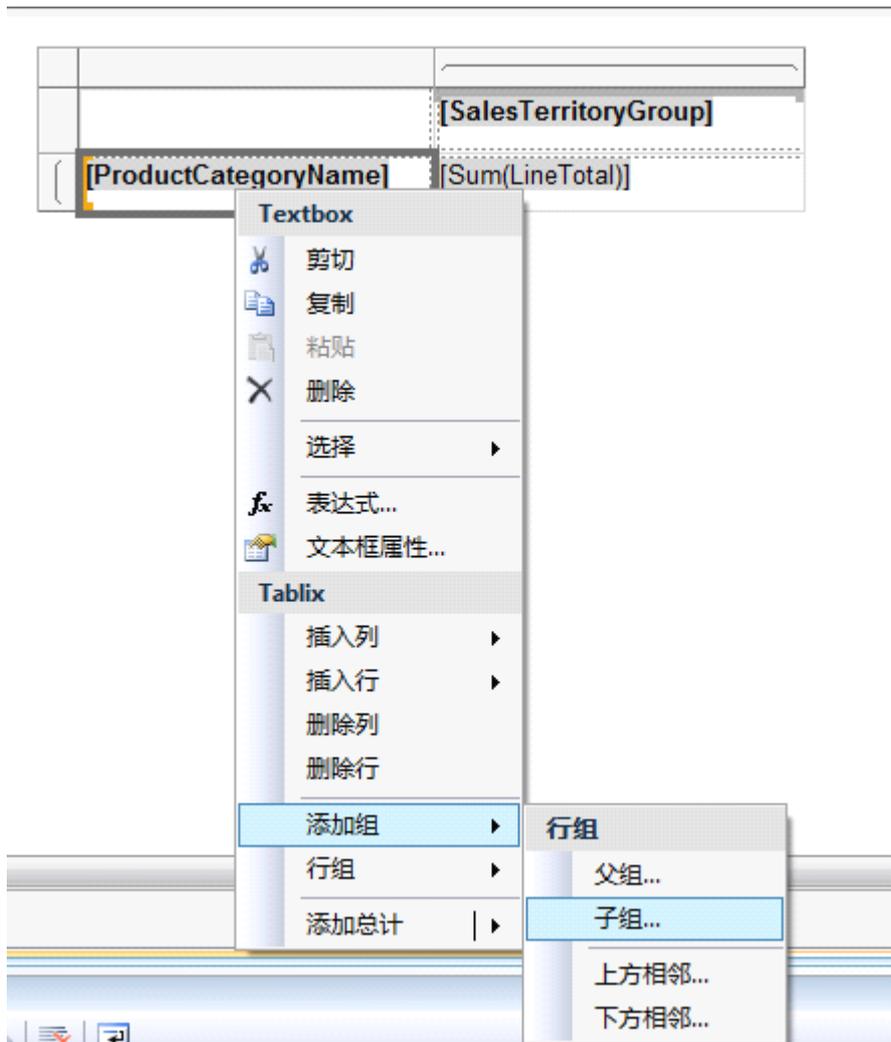
然后在工作区拖入矩阵。同时把数据集的 SalesTerritoryGroup 字段拖入列中，把 ProductCategoryName 填充到行，再把 LineTotal 填充到数据区，如图：

	[SalesTerritoryGroup]
[ProductCategoryName]	[Sum(LineTotal)]

点击预览将看到如下效果，是不是和表的列组功能相同~~

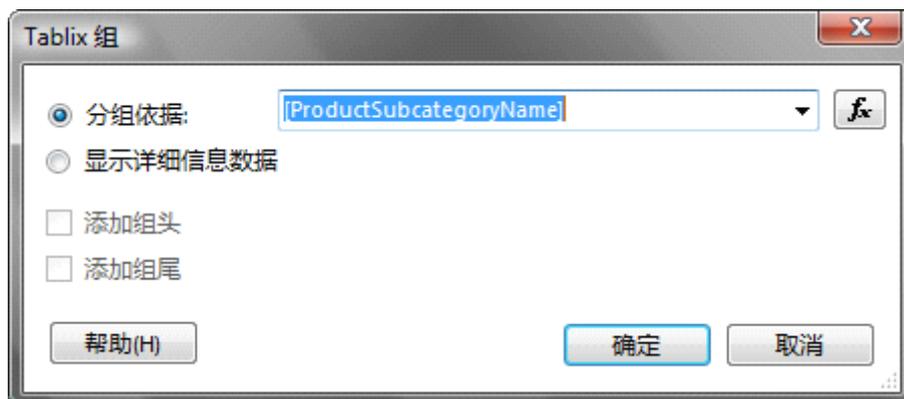
	Europe	North America	Pacific
Bikes	8633890.334276	18965954.898896	4373449.387200
Accessories	232200.341245	458240.146356	130678.536108
Clothing	204799.874167	556250.296600	85336.159786
Components	773200.912856	2281506.133668	116644.146000

上图中计算出了每种类型产品在各个地区的销售额。可是这时老板又有新的需求，需要既看到大的分类销售额，同时看到子分类的销售额。那我们该如何实现呢？OK，继续看~~



还记得我们在上一篇中，总是在添加“父组”，一直想点击“子组”，可一直没用上，现在就用上了。

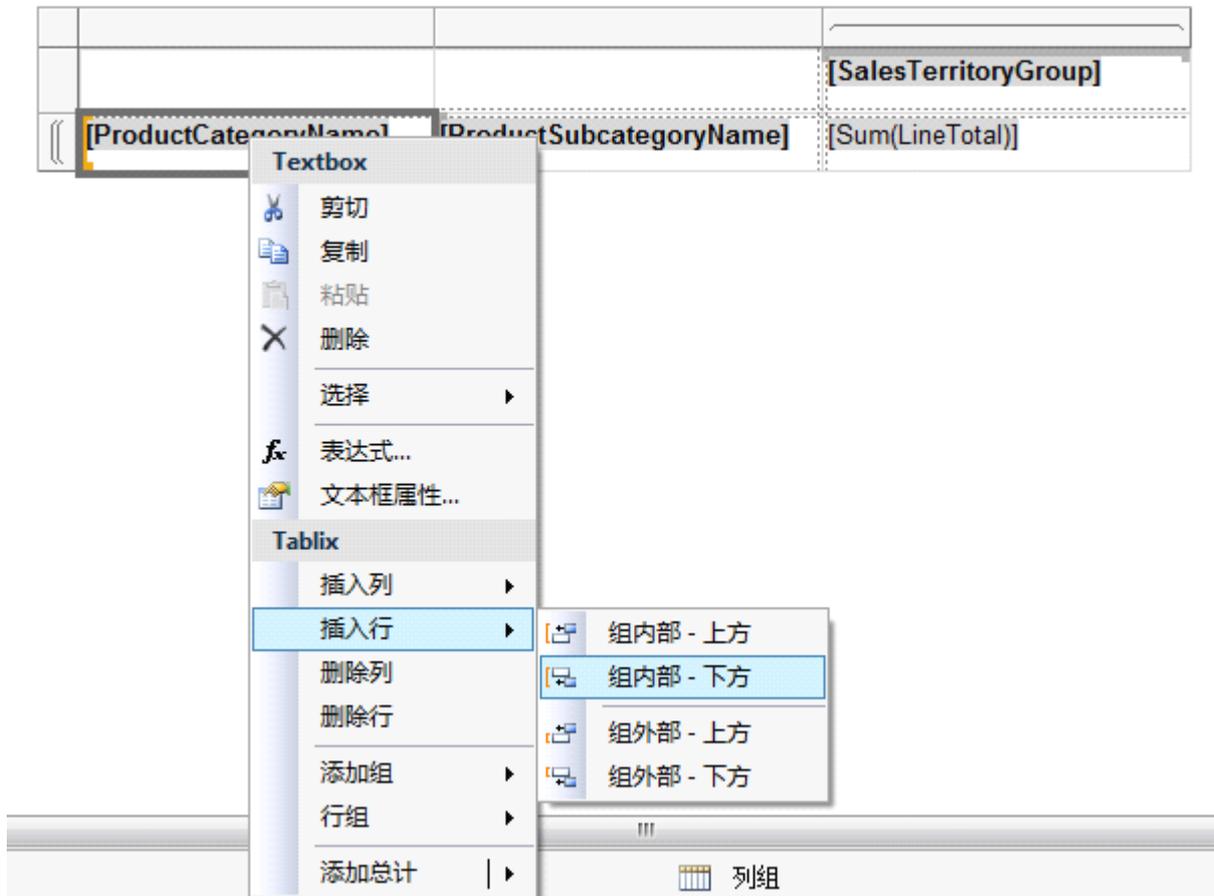
点击“子组”进入设置：设置方式几乎相同，太简单了。



点击确定再来看看工作区的矩阵发生了什么变化》

		[SalesTerritoryGroup]
[ProductCategoryName]	[ProductSubcategoryName]	[Sum(LineTotal)]

耶，已经按子类分组了，可是老板的要求是，既可以看到子类的销售额又可以看到大类的销售额。那怎么办？不用慌，我们只需要把大类的分组加上组头或组尾，然后汇总就可以了。我们这里添加组尾，来看看怎么添加。



右键 Product Category Name 组，在菜单中选择，插入行->“组内部-下方”。然后可以看到工作区的变化：

		[SalesTerritoryGroup]
[ProductCategoryName]	[ProductSubcategoryName]	[Sum(LineTotal)]
		[Sum(LineTotal)]

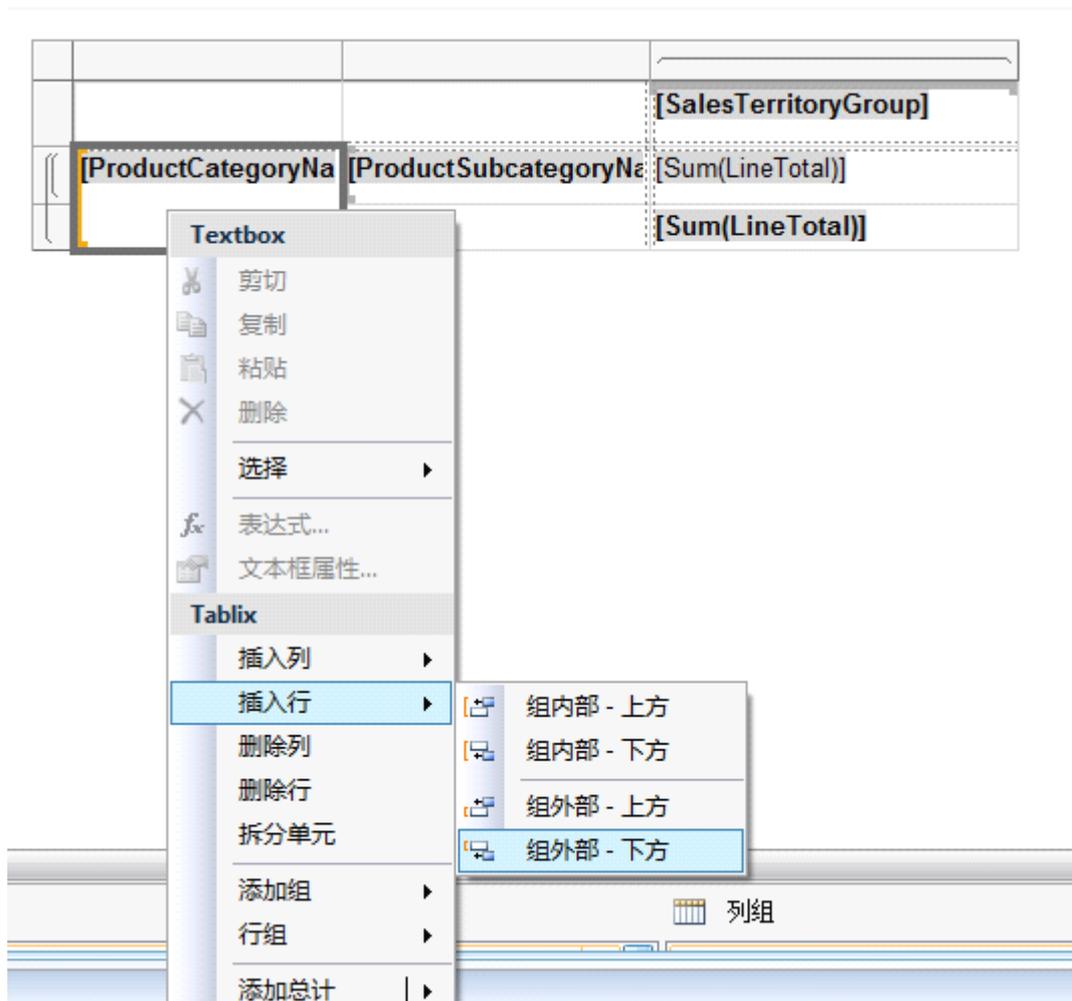
来看看效果~~

		Europe	North America	Pacific
Bikes	Mountain Bikes	2815609.917876	7042790.358160	1072537.953200
	Road Bikes	2464403.121569	6437672.860786	1545891.670000
	Touring Bikes	3353877.294831	5485491.679950	1755019.764000
	Total:	8633890.334276	18965954.898896	4373449.387200
Accessories	Bike Racks	38236.620000	97584.780000	13536.000000
	Bike Stands	8586.000000	15741.000000	8268.000000

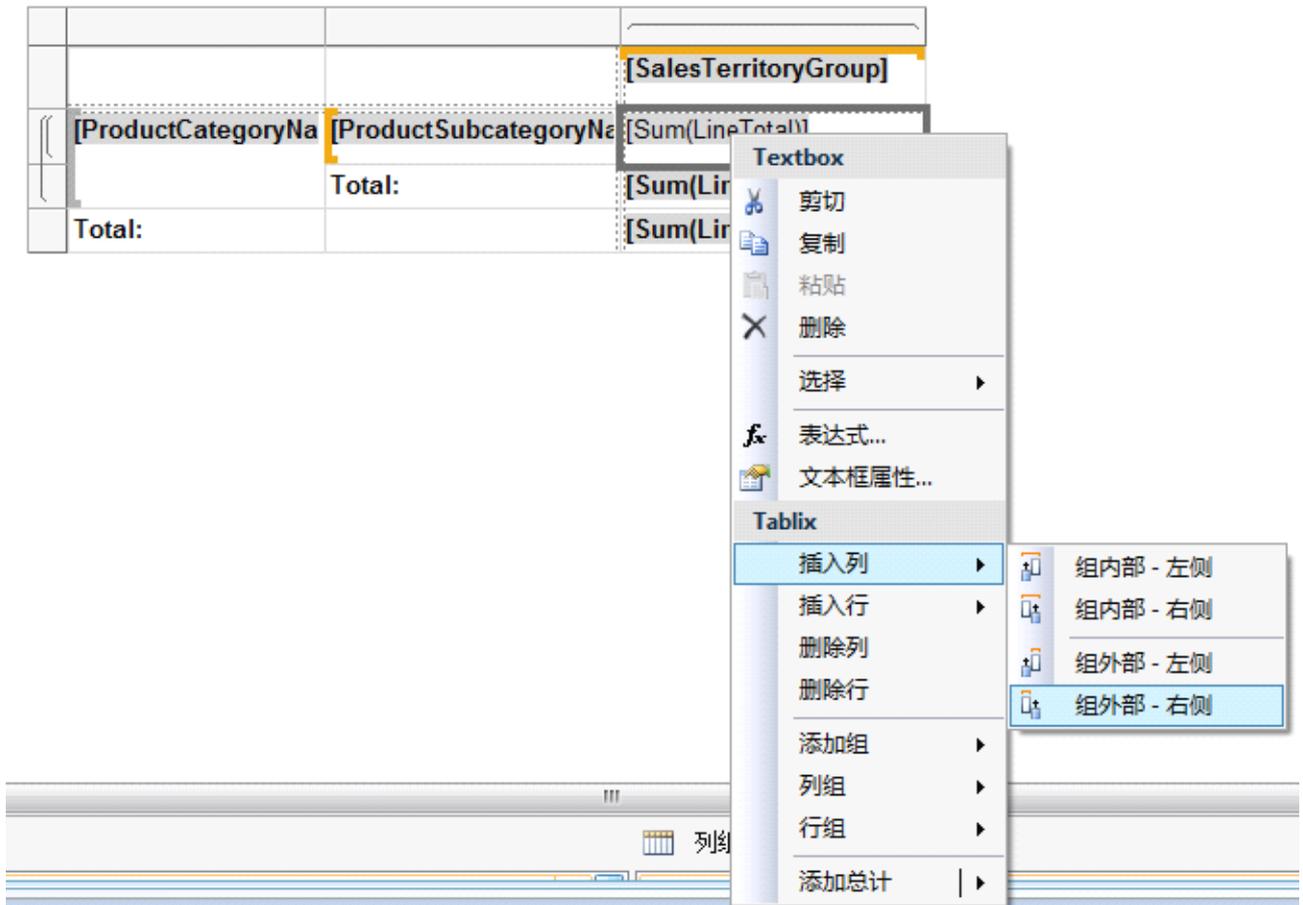
OK, 符合老板的要求。交工去~~, 老板看了报表后, 皱起眉头说: “我还想知道每个地区的总销售额及每个分类总销售额。” 这老板还真烦呀, 不会一起提出来呀。净折腾我们员工~~

继续完成吧~~

右键 Product Category Name 组, 在菜单中选择, 插入行->“组外部-下方” 来为每个地区汇总所有分类。



右键 **SalesTerritoryGroup** 组，在菜单中选择，插入列->“组外部-右方” 来为每个分类汇总所有地区。



添加后的设计界面如下：

		[SalesTerritoryGroup]	Total
[ProductCategoryName]	[ProductSubcategoryName]	[Sum(LineTotal)]	[Sum(LineTotal)]
	Total:	[Sum(LineTotal)]	[Sum(LineTotal)]
Total:		[Sum(LineTotal)]	[Sum(LineTotal)]

看看效果：

		Europe	North America	Pacific	Total
Bikes	Mountain Bikes	2815609.917876	7042790.358160	1072537.953200	10930938.229236
	Road Bikes	2464403.121569	6437672.860786	1545891.670000	10447967.652355
	Touring Bikes	3353877.294831	5485491.679950	1755019.764000	10594388.738781
	Total:	8633890.334276	18965954.898896	4373449.387200	31973294.620372

Components	Road Frames	117366.551600	448796.694000		566163.245600
	Saddles	6935.832000	19995.067200	1307.952000	28238.851200
	Touring Frames	287332.655328	545621.118000	69432.612000	902386.385328
	Wheels	83.298100			83.298100
	Total:	773200.912856	2281506.133668	116644.146000	3171351.192524
Total:		9844091.462544	22261951.475520	4706108.229094	36812151.167158

两个要求都完成了，这下应该满足老板的需求了。可是老板就是老板，总有提不完的需求～～。

这不，又来了，老板提出三条需求：

- 1: 计算出北美地区的每个国家的销售额。
- 2: 计算出每种分类，子类，每年的销售额。
- 3: 计算出每个子类在总类中占的销售百分比。

哇～，这么多需求，而且要在一张报表上完成，也太难了吧～～，我不入地狱谁入地狱～～。

先理清需求，再下手～这叫上有政策，下有对策（好像不适用）

- 1: 需要计算北美地区的国家的销售额，所以需要修改 SQL 脚本，加上地区条件。再添加子组列。
- 2: 计算出每种分类，子类，每年的销售额，看样还的加列。
- 3: 这个么，看样还的写表达式来计算了～

OK,先修改 SQL 脚本：如下脚本中的参数，在实际应用中可以提供给用户自己选择，这里为了方便，在此写固定值。

```
SELECT year(OrderDate) as Oyear,PC.Name AS ProductCategoryName,
```

```
PSC.Name as ProductSubcategoryName,CountryRegionCode,
ST.[Group] AS SalesTerritoryGroup,LineTotal
```

```
FROM Sales.SalesOrderHeader O
INNER JOIN Sales.SalesOrderDetail OD
ON O.SalesOrderID=OD.SalesOrderID
INNER JOIN Production.Product P
ON OD.ProductID=P.ProductID
INNER JOIN Production.ProductSubcategory PSC
ON P.ProductSubcategoryID=PSC.ProductSubcategoryID
INNER JOIN Production.ProductCategory PC
ON PSC.ProductCategoryID=PC.ProductCategoryID
INNER JOIN Sales.SalesTerritory ST
```

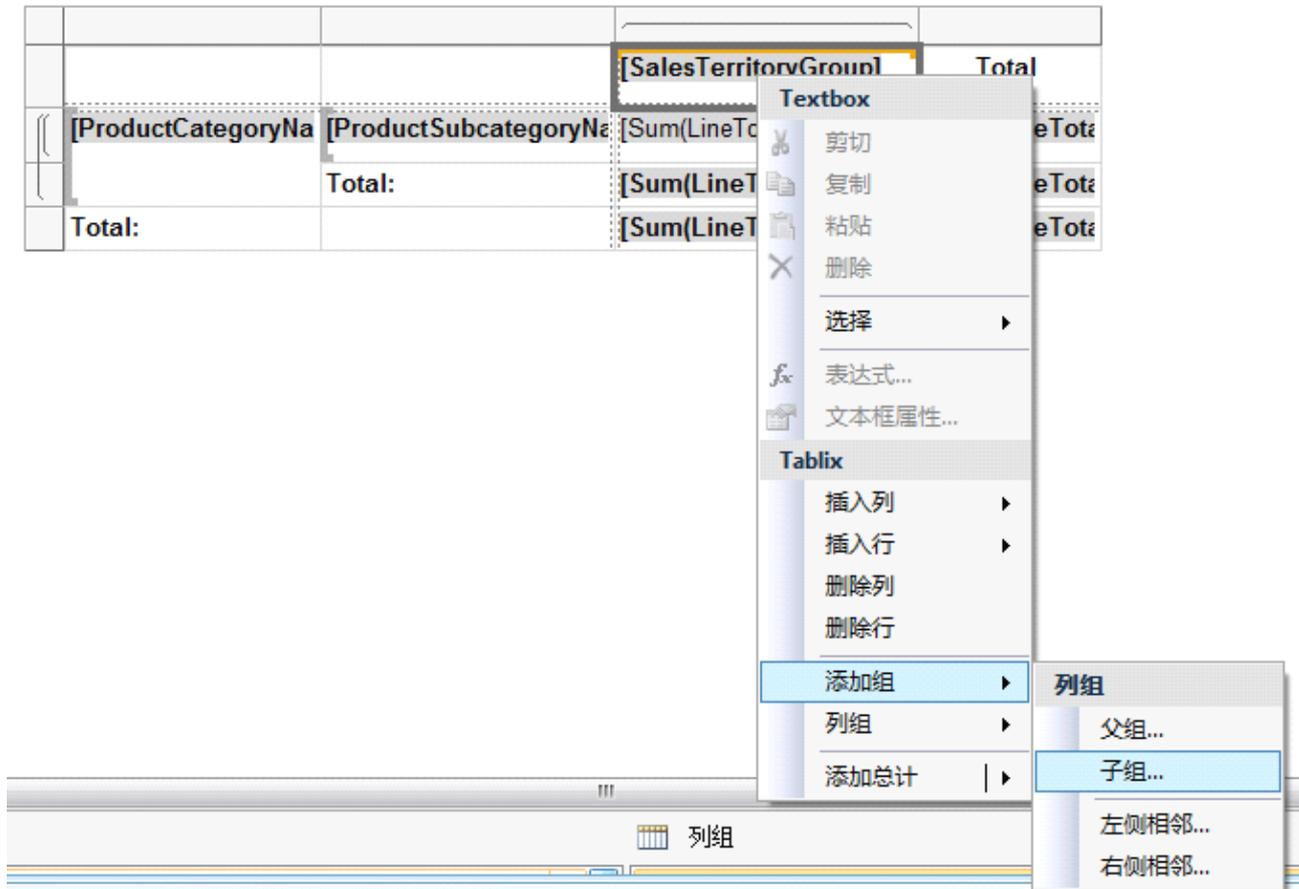
ON O.TerritoryID=ST.TerritoryID

where OrderDate>'20031001' AND OrderDate<'20041001'

and ST.[Group]='North America'

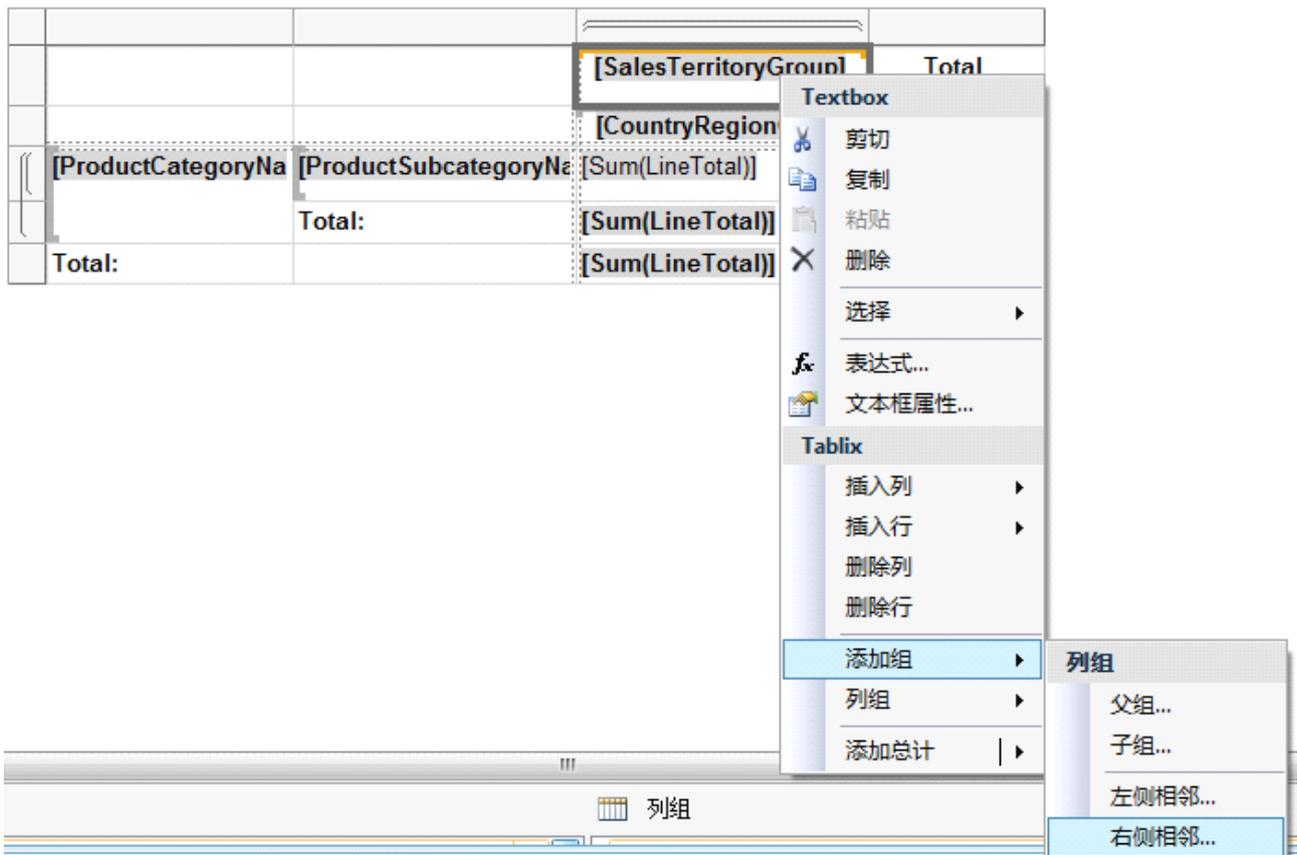
再来添加子组:

右键 **SalesTerritoryGroup** 组，在菜单中选择，添加组->“子组”。



在弹出的设置窗口，选择 **CountryRegionCode** 作为分组依据。这样第一个需求就完成了。

下面再来完成“计算出每种分类，子类，每年的销售额”需求。我们需要添加列组，我们希望显示在地区信息的右边，所有在下图中，我们选择“右侧相邻”。



在设置窗口中，我们设置 Oyear 为分组依据。填充上数据（**[Sum(LineTotal)]**）。

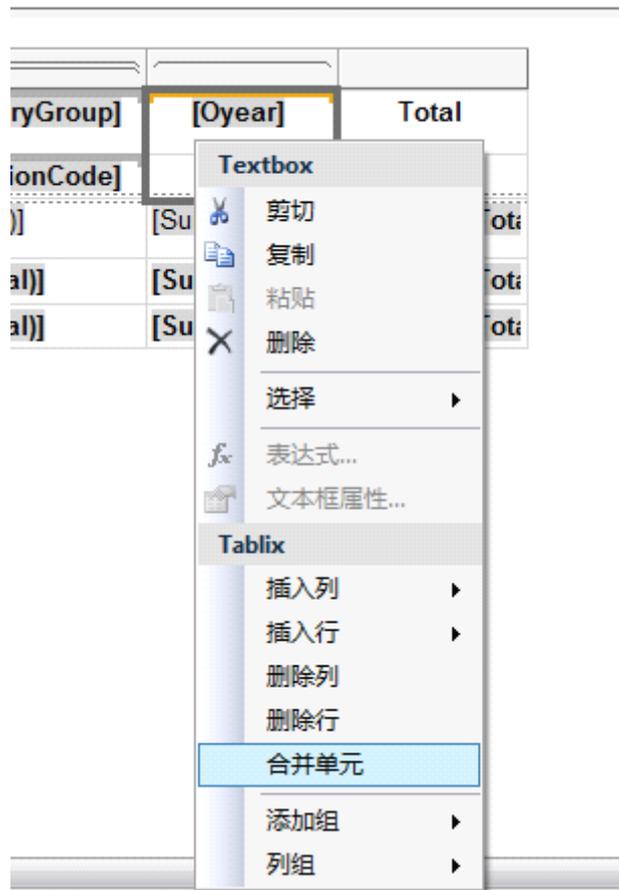
这时我们可以看到工作区的矩阵已经变化成如下：

		[SalesTerritoryGroup]	[Oyear]	Total
		[CountryRegionCode]		
[ProductCategoryName]	[ProductSubcategoryName]	[Sum(LineTotal)]	[Sum(LineTotal)]	[Sum(LineTotal)]
	Total:	[Sum(LineTotal)]	[Sum(LineTotal)]	[Sum(LineTotal)]
Total:		[Sum(LineTotal)]	[Sum(LineTotal)]	[Sum(LineTotal)]

运行一下看看效果吧～

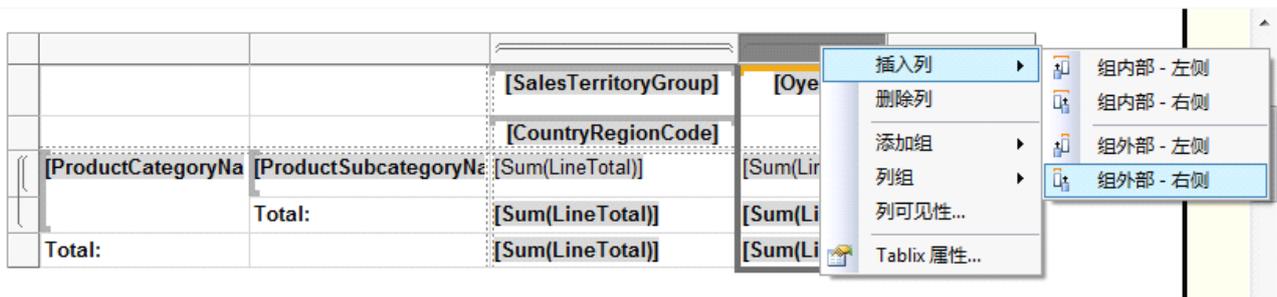
		North America		2003	2004	Total
		CA	US			
Accessories	Bike Racks	30813.684000	66771.096000	31236.900000	66347.880000	97584.780000
	Bike Stands	4293.000000	11448.000000	5724.000000	10017.000000	15741.000000
	Bottles and Cages	7099.040960	19826.676168	7722.861484	19202.855644	26925.717128
	Cleaners	2280.721440	5558.611380	2429.427780	5409.905040	7839.332820
	Fenders	7231.420000	18133.500000	7605.080000	17759.840000	25364.920000
	Helmets	39430.290483	98911.817404	42160.742131	96181.365756	138342.107887

基本上符合老板的要求了，但是年份列，和 Total 希望垂直居中，所有这里我们需要“合并单元”。



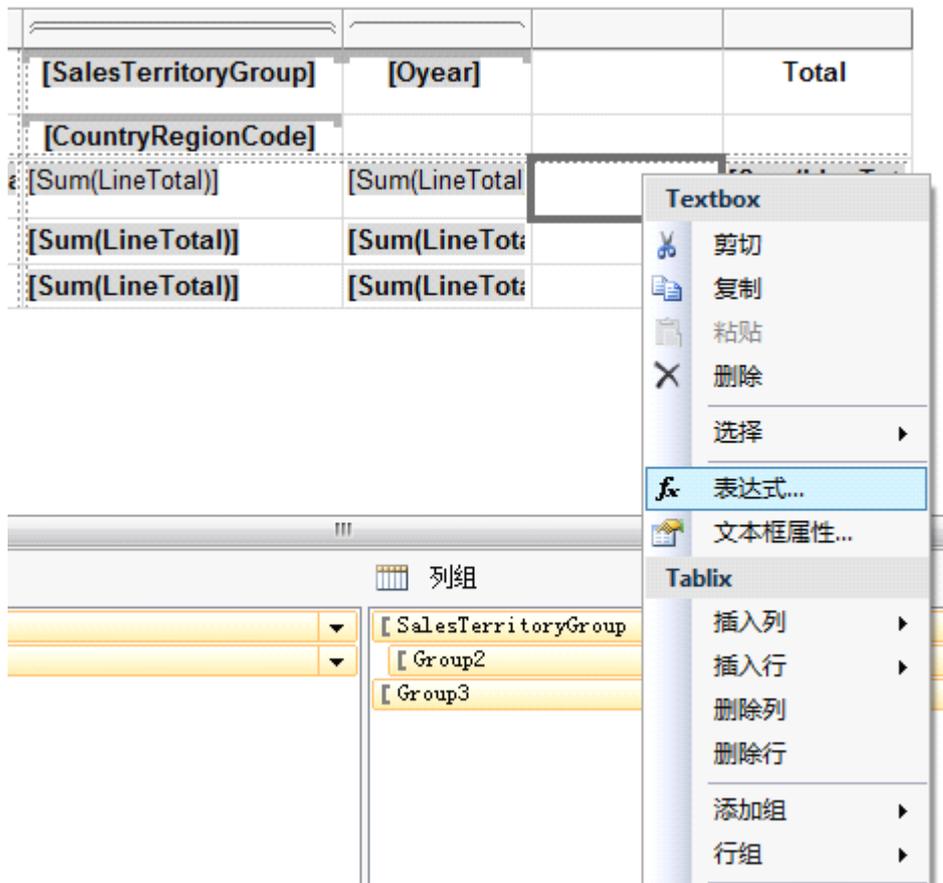
合并后的，这样就达到预期的效果了。

下面再来看看最后一个需求怎么完成。我们也需要添加一个组外列，来存放每个子类占该类的比例。所以如下图插入列。



然后我们填入具体的表达式：

`=SUM(Fields!LineTotal.Value)/SUM(Fields!LineTotal.Value,"ProductCategoryName")`



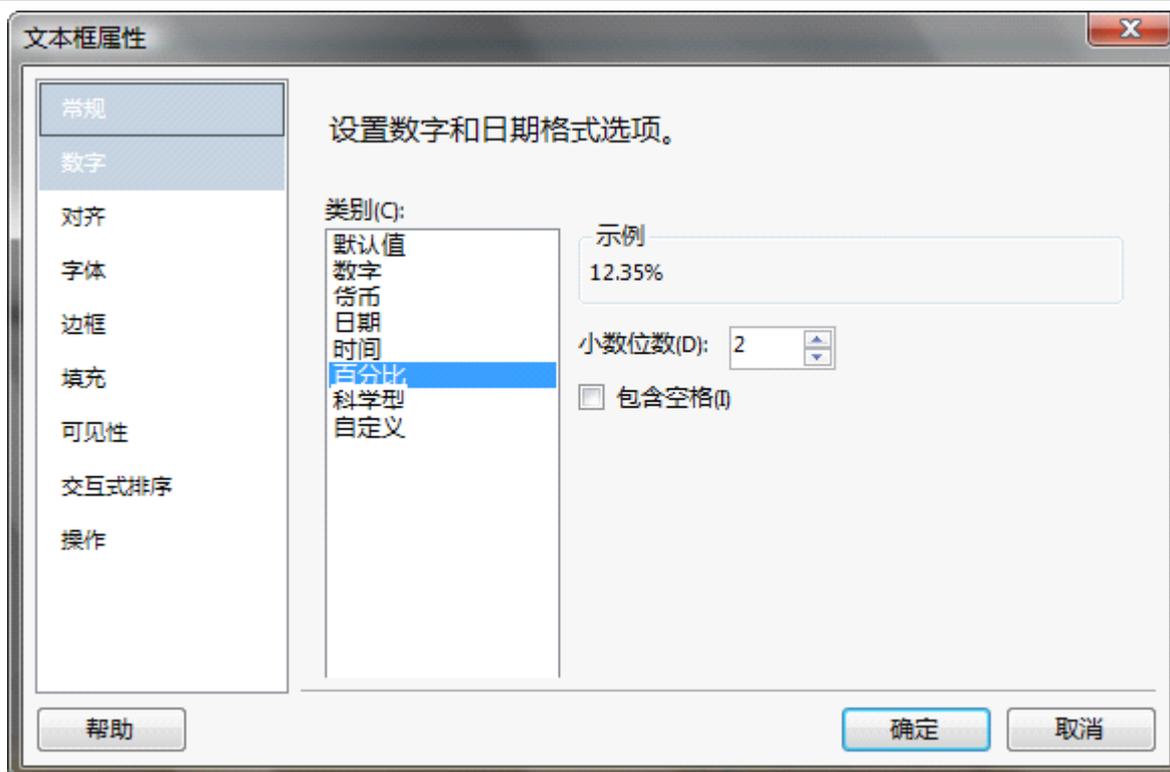
设置好列标题，工作区应该如下图：

		[SalesTerritoryGroup]	[Oyear]	%	Total
		[CountryRegionCode]			
[ProductCategoryName]	[ProductSubcategoryName]	[Sum(LineTotal)]	[Sum(LineTotal)]	«Expr»	[Sum(LineTotal)]
	Total:	[Sum(LineTotal)]	[Sum(LineTotal)]		[Sum(LineTotal)]
Total:		[Sum(LineTotal)]	[Sum(LineTotal)]		[Sum(LineTotal)]

迫不及待了，赶紧看看效果吧～～。

		North America		2003	2004	%	Total
		CA	US				
Accessories	Bike Racks	30813.684000	66771.096000	31236.900000	66347.880000	0.21295554476 405004039955 54454	97584.780000
	Bike Stands	4293.000000	11448.000000	5724.000000	10017.000000	0.03435098414 046649165914 40004	15741.000000
	Bottles and Cages	7099.040960	19826.676168	7722.861484	19202.855644	0.05875896588 746680467420 63877	26925.717128
	Cleaners	2280.721440	5558.611380	2429.427780	5409.905040	0.017110747712	7839.332820

耶～，基本功能完成了。不过百分比显示的是小数，那咋办呢？不用急。我们来设置一下：



在文本框属性中，我们设置成数字，百分比就 OK 了。

再来看看效果：

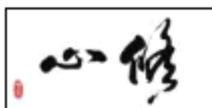
		North America		2003	2004	%	Total
		CA	US				
Accessories	Bike Racks	30813.684000	66771.096000	31236.900000	66347.880000	21.30%	97584.780000
	Bike Stands	4293.000000	11448.000000	5724.000000	10017.000000	3.44%	15741.000000
	Bottles and Cages	7099.040960	19826.676168	7722.861484	19202.855644	5.88%	26925.717128
	Cleaners	2280.721440	5558.611380	2429.427780	5409.905040	1.71%	7839.332820
	Fenders	7231.420000	18133.500000	7605.080000	17759.840000	5.54%	25364.920000

没错，我们终于完成这个庞然大物了。这下看老板还有什么说的~~肯定给你涨工资 o(∩_∩)o...

休息一下 o(∩_∩)o...我们再来以一个实例讲解一下“矩形”，“列表”，“图像”，“子报表”控件的使用。

“列表，矩形，图像”的综合使用

今天老板需要我们完成什么样的报表呢？只见老板笑眯眯的说：“我的要求很简单，只想根据订单号，查询订单的下单日期，发货日期，及总金额情况，还有相关的商品信息。来看看我的 Sample，只需要按照我的 Sample 完成就 OK 了。完成了年底就有年终奖~~”。



S043659订单信息

订单日期: 2001/7/1 0:00:00 订单发送给客户的日期: 2001/7/8 0:00:00

消费情况: (订单金额+税额+运费)

24843.9362 + 1971.5149 + 616.0984 = 27231.5495

订单相关地址:

BillTo: 42525 Austell Road ShipTo: 42525 Austell Road

订单明细:

Product Name	Unit Price	Unit Price Discount	Order Qty	Line Total
Mountain-100 Black, 42	2024.9940	0.0000	1	2024.994000
Mountain-100 Black, 44	2024.9940	0.0000	3	6074.982000
Mountain-100 Black, 48	2024.9940	0.0000	1	2024.994000
Mountain-100 Silver, 38	2039.9940	0.0000	1	2039.994000

看了 Sample 后，大家就知道老板是别有用心呀，明摆着不想发年终奖，这么复杂的报表如何才能完成？

前面我们学到的可都是有固定格式的，行行列列的，现在这么任意摆放，无规律可循的报表该如何下手？

不要慌，我们静下心来，仔细的研究一下 Sample 到底有哪些东西。

“图像，线条，文本框，表”，还有不知道啥东西可以任意摆放文本框？

OK,下面我们就来学习一下这个自由主义者“列表”及“矩形”。先简单的来理解一下两者：列表：用于创建与网页上的表单类似的自由格式布局。矩形：用来放置其他报表项的容器。

首先新建报表，新建数据集，这里用到两个数据集，一个订单信息，一个订单明细信息。

SQL脚本如下：两个数据集的参数都为：[@SalesOrderNumber](#)

OrderInfo:

```
select OrderDate,SalesOrderNumber,a.AddressLine1 as BillToAddress1,
a.AddressLine2 as BillToAddress2,a1.AddressLine1 as ShipToAddress1,
a1.AddressLine2 as ShipToAddress2,SubTotal,TaxAmt,Freight,TotalDue,
DueDate,ShipDate
from Sales.SalesOrderHeader o
inner join Person.[Address] a
on o.BillToAddressID=a.AddressID
```

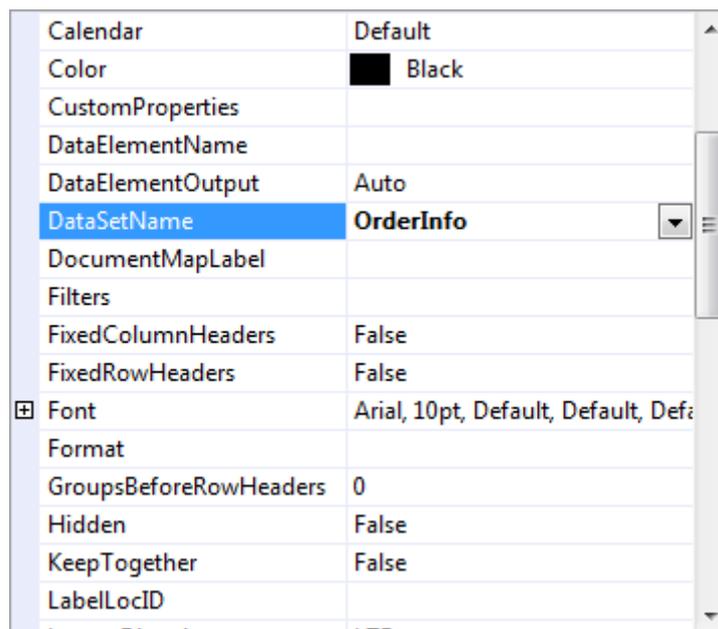
```
inner join Person.[Address] a1
on o.ShipToAddressID=a1.AddressID
where SalesOrderNumber=@SalesOrderNumber
```

OrderDetail:

```
SELECT OrderDate,SalesOrderNumber,P.Name AS ProductName,
UnitPrice,UnitPriceDiscount,OrderQty,LineTotal
FROM Sales.SalesOrderHeader O
INNER JOIN Sales.SalesOrderDetail OD
ON O.SalesOrderID=OD.SalesOrderID
INNER JOIN Production.Product P
ON OD.ProductID=P.ProductID
where SalesOrderNumber=@SalesOrderNumber
```

“列表”使用

建好数据集后，我们把 List 控件拖入工作区。调整适当的大小。绑定 List 的数据集为 **OrderInfo**。如图：



初步工作完成后，我们来摆放 List 中需要的元素，按照前面章节学习的方法，拖入文本框作为标题。

如图：



通过右键文本框，再弹出的菜单中选择“表达式”。我们希望标题显示为“订单号”+订单信息字样，填写表达式内容为：`=Fields!SalesOrderNumber.Value+"订单信息"`。

然后按照相同的方式来完成，订单日期，发货日期，消费情况的设置。设置好后工作区的变化如下：



“矩形”使用

下面我们来设置订单的相关地址，我们希望地址显示在一个区块内，并且在地址周围添加边框。所以需要拖入“矩形”控件。并设置显示边框。



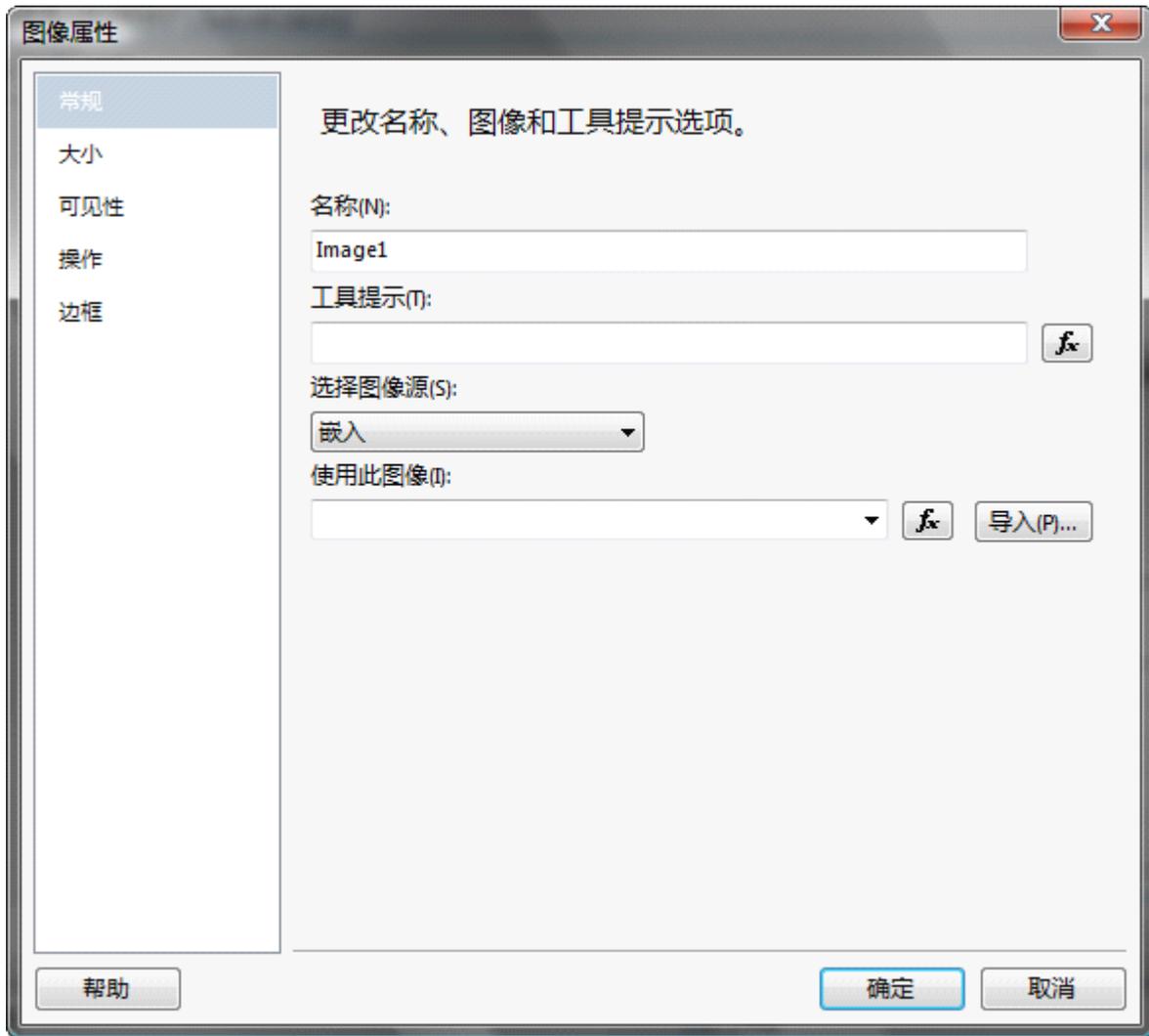
我们在矩形区块放入文本框来显示地址内容。如下图：

The screenshot shows a report design view with a grey header bar. The main content area is a white rectangle with a thin border. At the top center is a dashed box containing the text «Expr». Below this, there are two rows of labels and data fields. The first row contains '订单日期:' followed by a dashed box with '[OrderDate]', and '订单发送给客户的日期:' followed by a dashed box with '[ShipDate]'. The second row contains '消费情况：（订单金额 + 税额 + 运费）' followed by a dashed box with '[SubTotal]', a '+' sign, a dashed box with '[TaxAmt]', another '+' sign, a dashed box with '[Freight]', an '=' sign, and a dashed box with '[TotalDue]'. A horizontal line separates this from the next section. Below the line is a dashed box containing the text '订单相关地址：'. Underneath this is a larger dashed box containing two columns of labels and data fields. The left column has 'BillTo:' followed by a dashed box with '[BillToAddress1]' and another dashed box with '[BillToAddress2]'. The right column has 'ShipTo:' followed by a dashed box with '[ShipToAddress1]' and another dashed box with '[ShipToAddress2]'. On the left side of the design view, there is a vertical grey bar with a yellow highlight and a small '≡' icon.

List 中的内容设置完了，不对，好像少了什么？对的，少了图片。我们来看看如何添加图片。

“图像”使用

首先从工具栏拖入“图片”控件到相应位置，会弹出如下窗口。



在此窗口，可以设置图像名称，大小，可见性，边框，操作等。

图像源：

“外部”：是对存储在 Web 服务器中的图像的 URL 引用

“嵌入”：对嵌入的图像数据引用，

“数据库”：是对数据库中的二进制图像数据引用。

Reporting Services 支持 .bmp、.jpeg、.gif 和 .png 文件。

在此我们选择“嵌入”，点击导入按钮来导入图片。调整图片的位置及大小。

可见如下效果：

心修

«Expr»

订单日期: [OrderDate] 订单发送给客户的日期: [ShipDate]

消费情况: (订单金额 + 税额 + 运费)

[SubTotal] + [TaxAmt] + [Freight] = [TotalDue]

订单相关地址:

BillTo: [BillToAddress1] ShipTo: [ShipToAddress1]
 [BillToAddress2] [ShipToAddress2]

这样，LIST 才算设置完成。下面我们在 LIST 的下面添加一个“表”控件来显示该订单的所买商品信息。添加“表”控件，绑定数据源为：OrderDetail，并填充字段。最终设计效果如下：

心修

«Expr»

订单日期: [OrderDate] 订单发送给客户的日期: [ShipDate]

消费情况: (订单金额 + 税额 + 运费)

[SubTotal] + [TaxAmt] + [Freight] = [TotalDue]

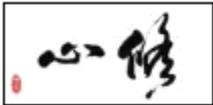
订单相关地址:

BillTo: [BillToAddress1] ShipTo: [ShipToAddress1]
 [BillToAddress2] [ShipToAddress2]

订单明细:

Product Name	Unit Price	Unit Price Discount	Order Qty	Line Total
[ProductName]	[UnitPrice]	[UnitPriceDiscount]	[OrderQty]	[LineTotal]

点击“预览”，填入订单号来看一下效果是否符合老板的要求。



S043659订单信息

订单日期: 2001/7/1 0:00:00 **订单发送给客户的日期:** 2001/7/8 0:00:00

消费情况: (订单金额+税额+运费)
 24643.9362 + 1971.5149 + 616.0984 = 27231.5495

订单相关地址:

BillTo: 42525 Austell Road	ShipTo: 42525 Austell Road
-----------------------------------	-----------------------------------

订单明细:

Product Name	Unit Price	Unit Price Discount	Order Qty	Line Total
Mountain-100 Black, 42	2024.9940	0.0000	1	2024.994000
Mountain-100 Black, 44	2024.9940	0.0000	3	6074.982000
Mountain-100 Black, 48	2024.9940	0.0000	1	2024.994000
Mountain-100 Silver, 38	2039.9940	0.0000	1	2039.994000

可见和前面的 Sample 一样。这下老板亏大了，拿着报表去领年终奖了~~

可是刚要开口，老板却先说了“我知道你完成的很好，可是你总不能让我一个一个订单输入来查看吧，何况我也记不得那么多订单号呀，所以得完善一下报表，可以从所有订单点击进去查看明细”。还有，还有我想看到每个商品的图片。

所以还得继续做~~

OK,我们先给商品加上图片。

修改 OrderDetail 数据集如下:

```

SELECT OrderDate,SalesOrderNumber,P.Name AS ProductName,
UnitPrice,UnitPriceDiscount,OrderQty,LineTotal,ThumbNailPhoto
FROM Sales.SalesOrderHeader O
INNER JOIN Sales.SalesOrderDetail OD
ON O.SalesOrderID=OD.SalesOrderID
INNER JOIN Production.Product P
ON OD.ProductID=P.ProductID
inner join Production.ProductProductPhoto ppp
  
```

```

on p.ProductID=ppp.ProductID
inner join Production.ProductPhoto pp
on pp.ProductPhotoID=pp.ProductPhotoID
where SalesOrderNumber=@SalesOrderNumber

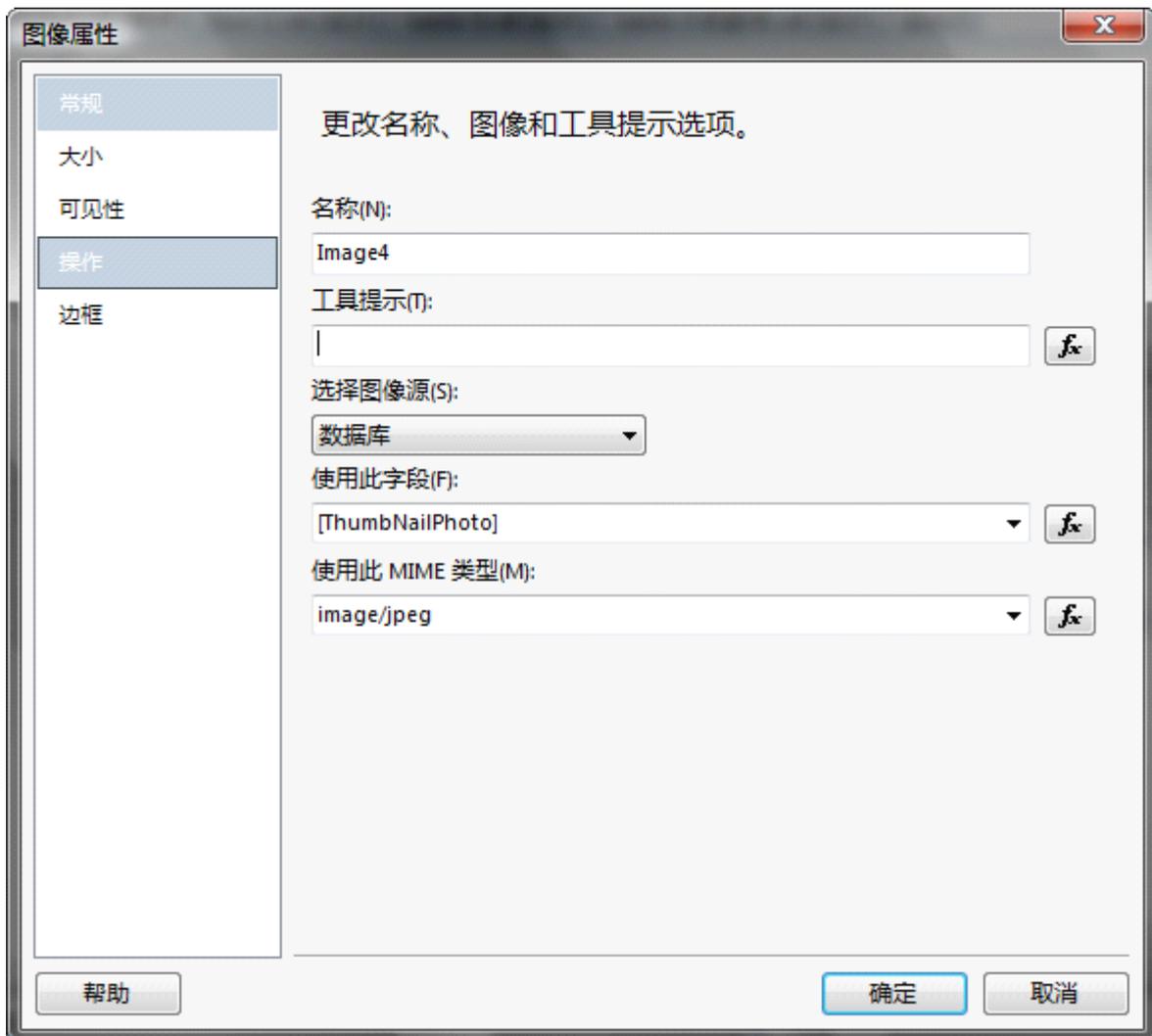
```

在 OrderDetail 报表的订单明细中增加一列来放置商品的图片。如下图:

订单明细:

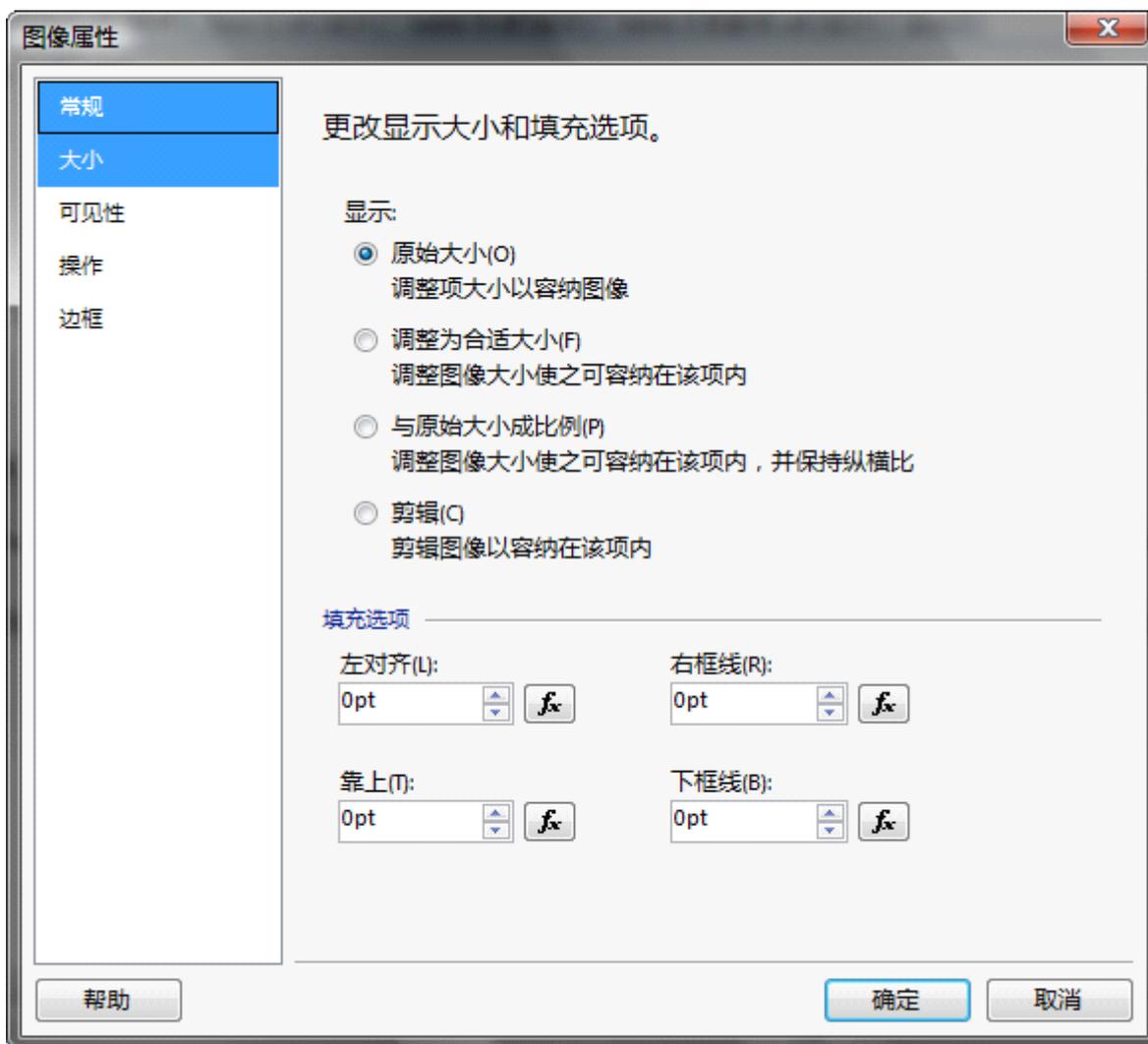
Product Name	Product Photo	Unit Price	Unit Price Discount	Order Qty	Line Total
[ProductName]		[UnitPrice]	[UnitPriceDiscount]	[OrderQty]	[LineTotal]

从工具栏拖入图像控件到该列将出现如下图的图像属性窗口:



我们设置“图像源”为数据库，字段为：**【ThumbNailPhoto】**，MIME 类型选择为：image/jpeg。

在“大小”选项卡设置图片为“原始大小”。



点击确定来看一下效果：

订单明细：

Product Name	Product Photo	Unit Price	Unit Price Discount	Order Qty	Line Total
Water Bottle - 30 oz.		4.9900	0.0000	1	4.990000

可见图片已经从数据库里读出来。这里也可以在列中先放入“矩形”控件再放入图像，达到的效果一样。

为报表添加链接

下面我们在来做一张所有订单的列表，可以点击订 单 ID 查看详细信息。

新建一张报表名称为：Order 的报表。数据集如下：

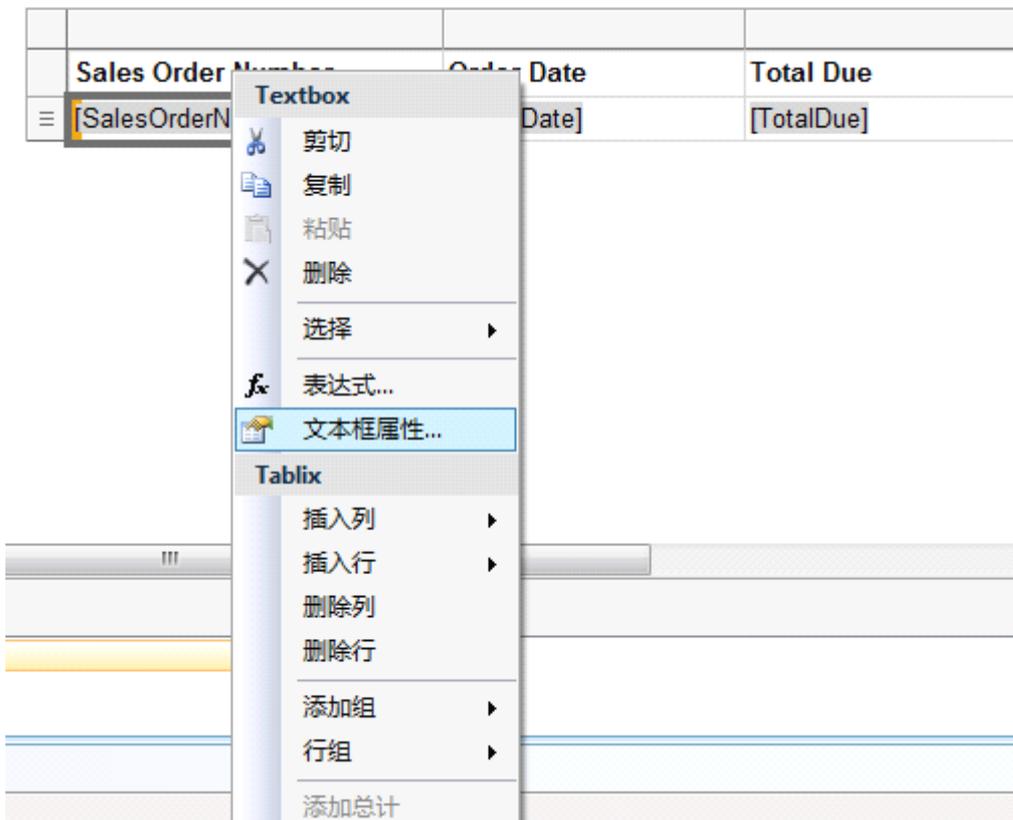
Order:

```
select SalesOrderNumber,OrderDate,TotalDue
from Sales.SalesOrderHeader
```

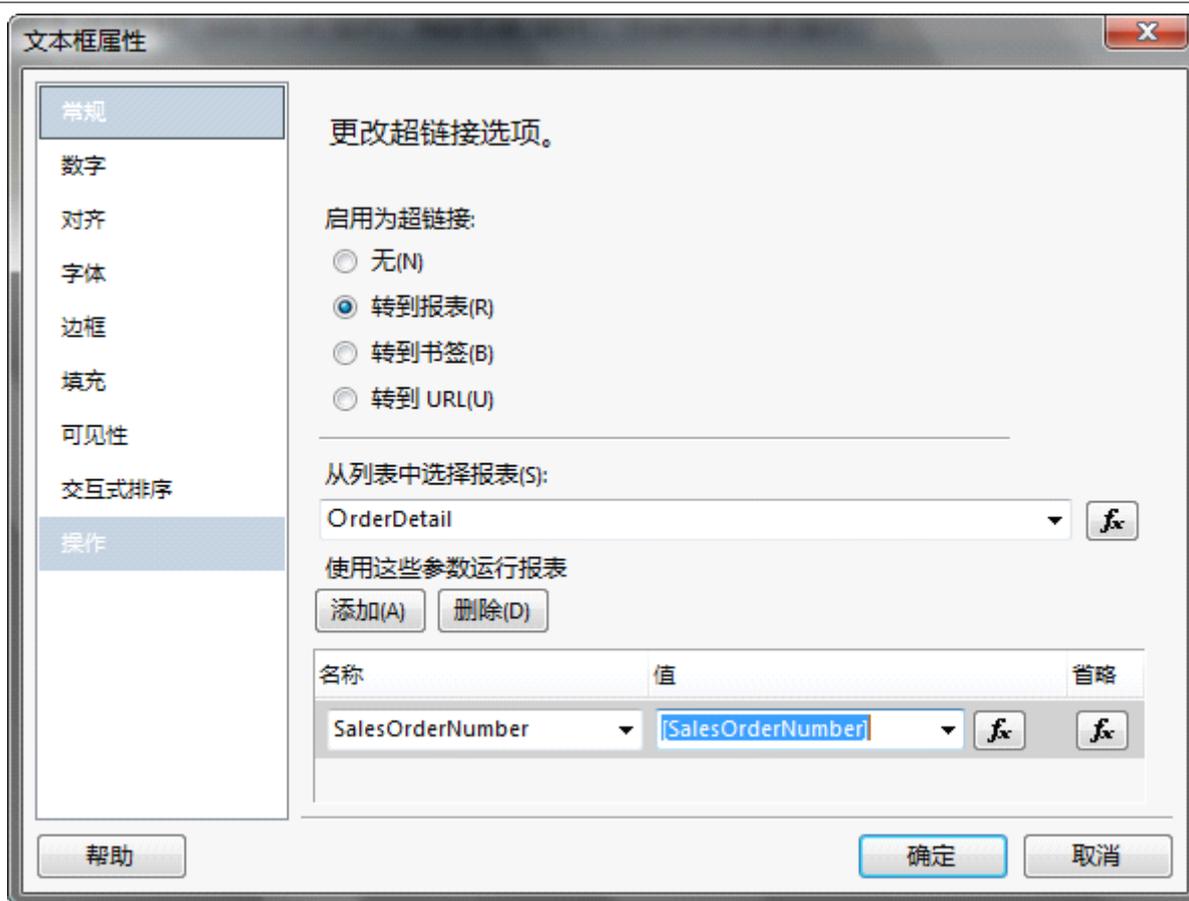
我们在工作区拖入“表”控件，绑定数据集并填充数据。如下图：

Sales Order Number	Order Date	Total Due
[SalesOrderNumber]	[OrderDate]	[TotalDue]

下面我们来为 Sales Order Number 添加链接跳转到我们前面完成的 OrderDetail 报表。右键订单号的文本框，在弹出的菜单中，选择文本框属性。



在“操作”选项卡，我们设置启用超链接功能，并选择转到报表，在下拉列表中选择“OrderDetail”报表。并添加 SalesOrderNumber 参数。同时在“字体”选项卡中设置颜色为“蓝色”，带下划线。



点击“确定”，我们来看一下效果：

Sales Order Number	Order Date	Total Due
SO43659	2001/7/1 0:00:00	27231.5495
SO43660	2001/7/1 0:00:00	1716.1794
SO43661	2001/7/1 0:00:00	43561.4424
SO43662	2001/7/1 0:00:00	38331.9613
SO43663	2001/7/1 0:00:00	556.2026

当我们把鼠标放在“订单号”上将出现可点状。点击将跳转到明细表。

好了，可以交差了～～，可是没想到这老板还真是懒呀。连点都不想点。只想订单的明细导出在一个文件中。通过书签可以查看到订单明细～～～汗～～。

好吧，这你也难不到我～～，我们程序员最大的能耐就是可以忍受老板所有的折磨～～

“子报表”

下面来介绍一下子报表，它可以帮我们完成老板的还不算太变态的要求～～。

首先看一下“子报表”的定义：

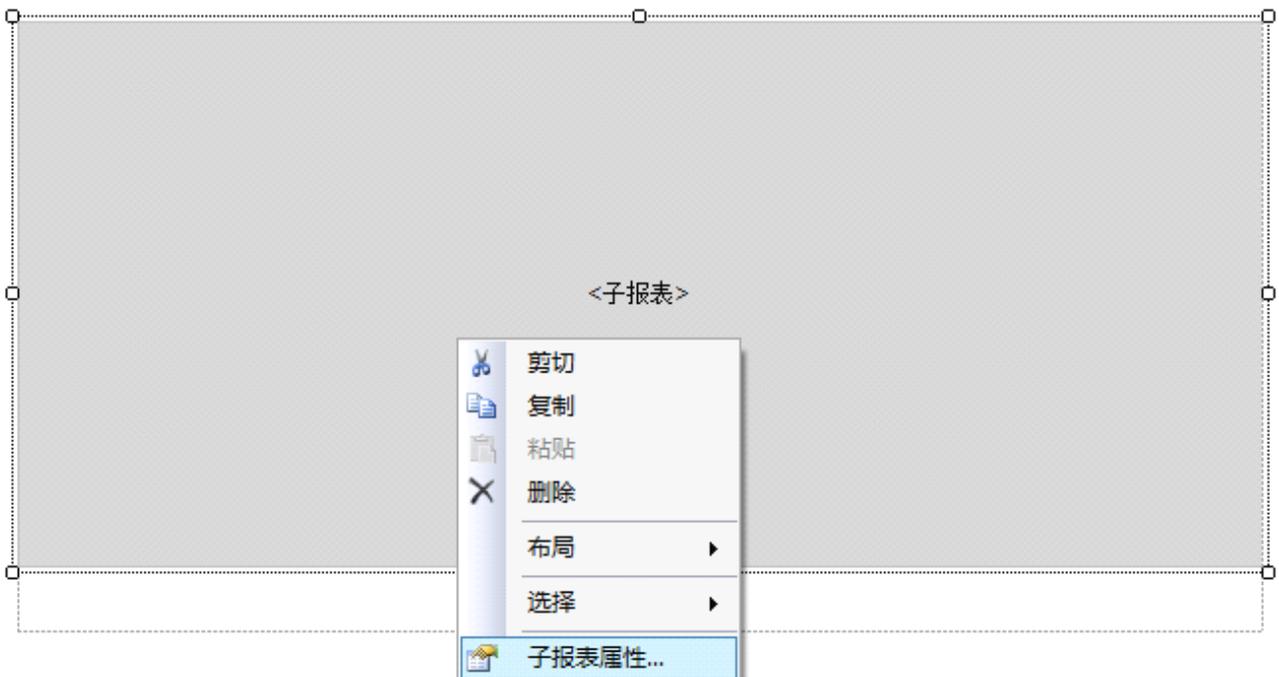
子报表：子报表显示保存到父报表的报表服务器中的报表。子报表可以引用完全格式化的独立报表，或嵌入主报表中的报表。定义子报表时，还可以定义用于筛选子报表数据的参数。

下面我们来修改 Order 报表，完成老板的需求。

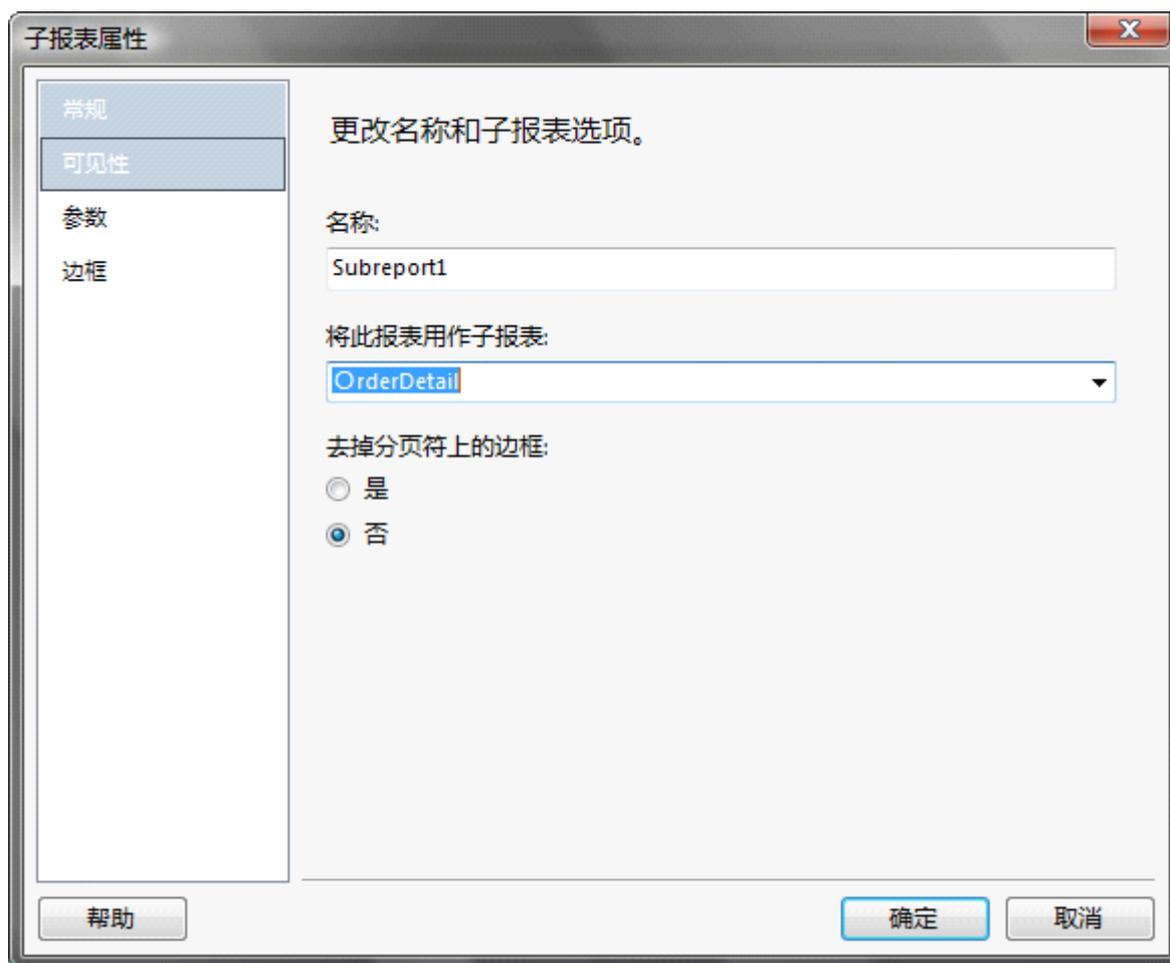
从报表中删除“表”控件，然后从工具栏拖入 List 控件，调整大小及位置。绑定数据集 **Order** 到列表控件。然后再从工具栏拖入子报表到 List 中。如下图：



大家可见 List 中放了一个子报表。我们来设置子报表的属性。

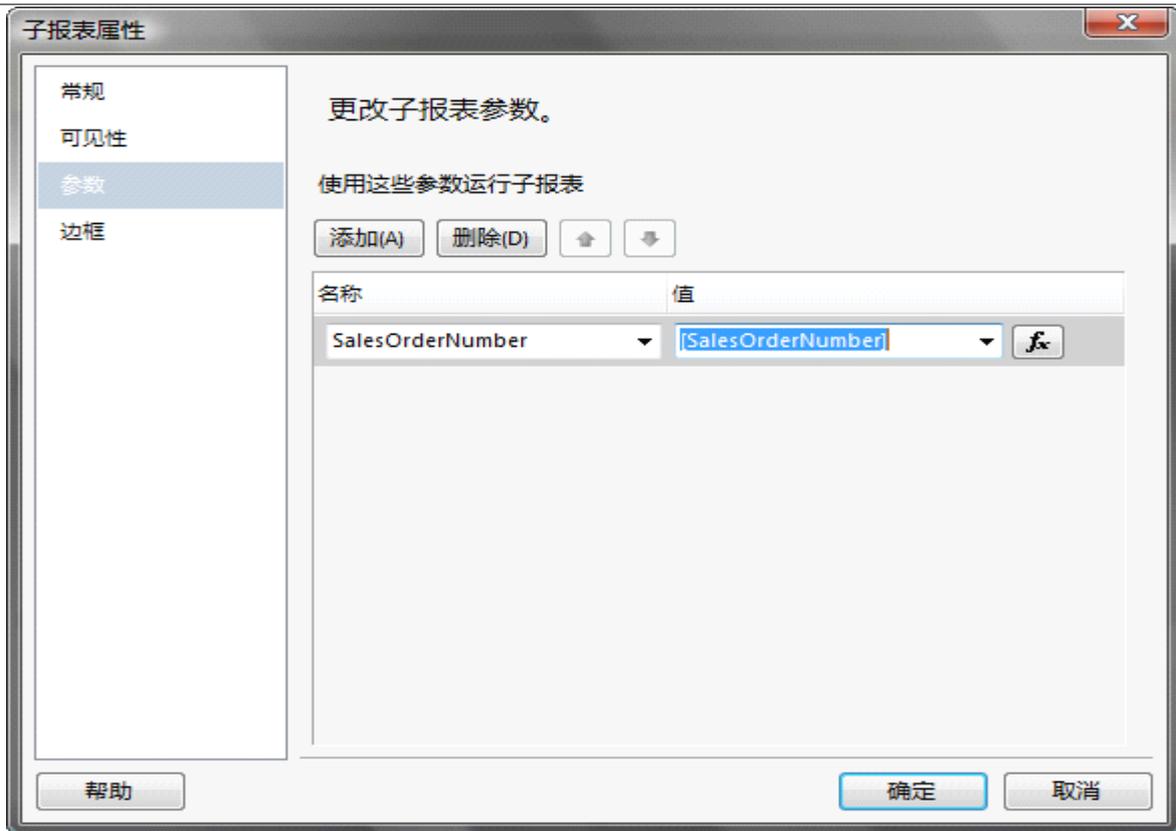


右键子报表在弹出的菜单中，选择“子报表属性”。



我们选择 OrderDetail 报表作为子报表。然后再选择“参数”选项卡，在此设置子报表的参数值。

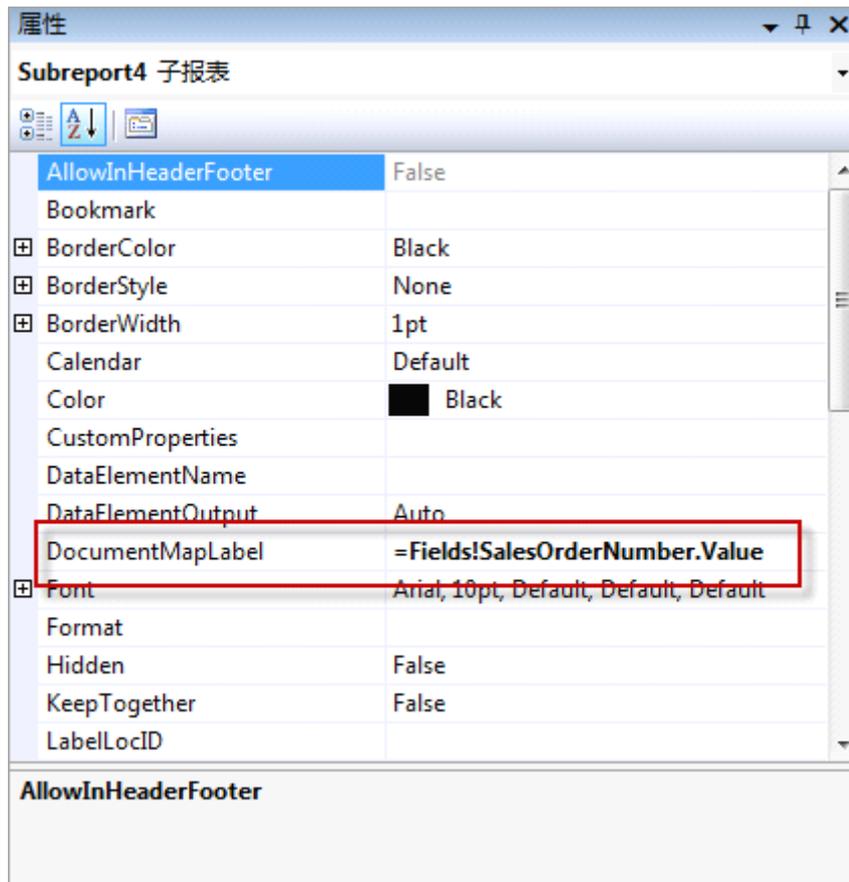
子报表的运行参数为：SalesOrderNumber



设置好后，工作区的界面如下：



下面我们来设置，让订单号以标签的方式显示。选择子报表控件，在属性窗口中设置“DocumentMapLabel”属性。设置值为：SalesOrderNumber 如图：



OK,一切就绪,来看看效果:

上面我们使用了列表对订单进行了循环,也可以用表控件,达到的效果相同。但是在2005中嵌套表格导出文件是有问题的。在2008中好像解决了这个问题。

OK,现在可以去和老板谈谈年终奖的问题了,这下老板应该没话可说了~~~。

本篇内容就讲到这里了。以上所讲的控件都是日常工作中经常用到的，所以得用心钻研，熟练掌握。根据业务需求的不同，选择合适的控件，大家在工作中发现有意思的使用方式也可以拿来让大家分享一下。希望以上内容没浪费大家的宝贵时间，最近身体不适，所以推出时间有点缓慢，敬请谅解！

在此感谢大家一如既往的支持~~下一篇将讲解“图表”及“仪表”控件，也是报表开发中最有意思的内容，同样是结合实例来讲解具体的使用。敬请期待~~~