

《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（二）开发篇

各位朋友，我们又相见了，经过上一篇枯燥乏味的配置过程后，将带领大家进入趣味十足的开发篇。虽然上一篇比较枯燥，但是大家经过自己动手去实验一下报表安装及配置，应该对 Reporting Services 有了一个初步的了解。本篇内容将以实例为主，来讲解各个报表控件的使用。

主要讲解 Tablix 控件的使用，Tablix 控件在2008中有了长足的改进，Tablix 结合了表和矩阵数据区的功能，可以合并多个列组合和行组合。能够同时支持动态行和动态列。闲话少扯~~直接进入主题吧。

新建项目

首先打开“开始”->“Microsoft SQL Server 2008”->“SQL Server Business Intelligence Development Studio”，这是我们开发报表的主要工具。见下图：



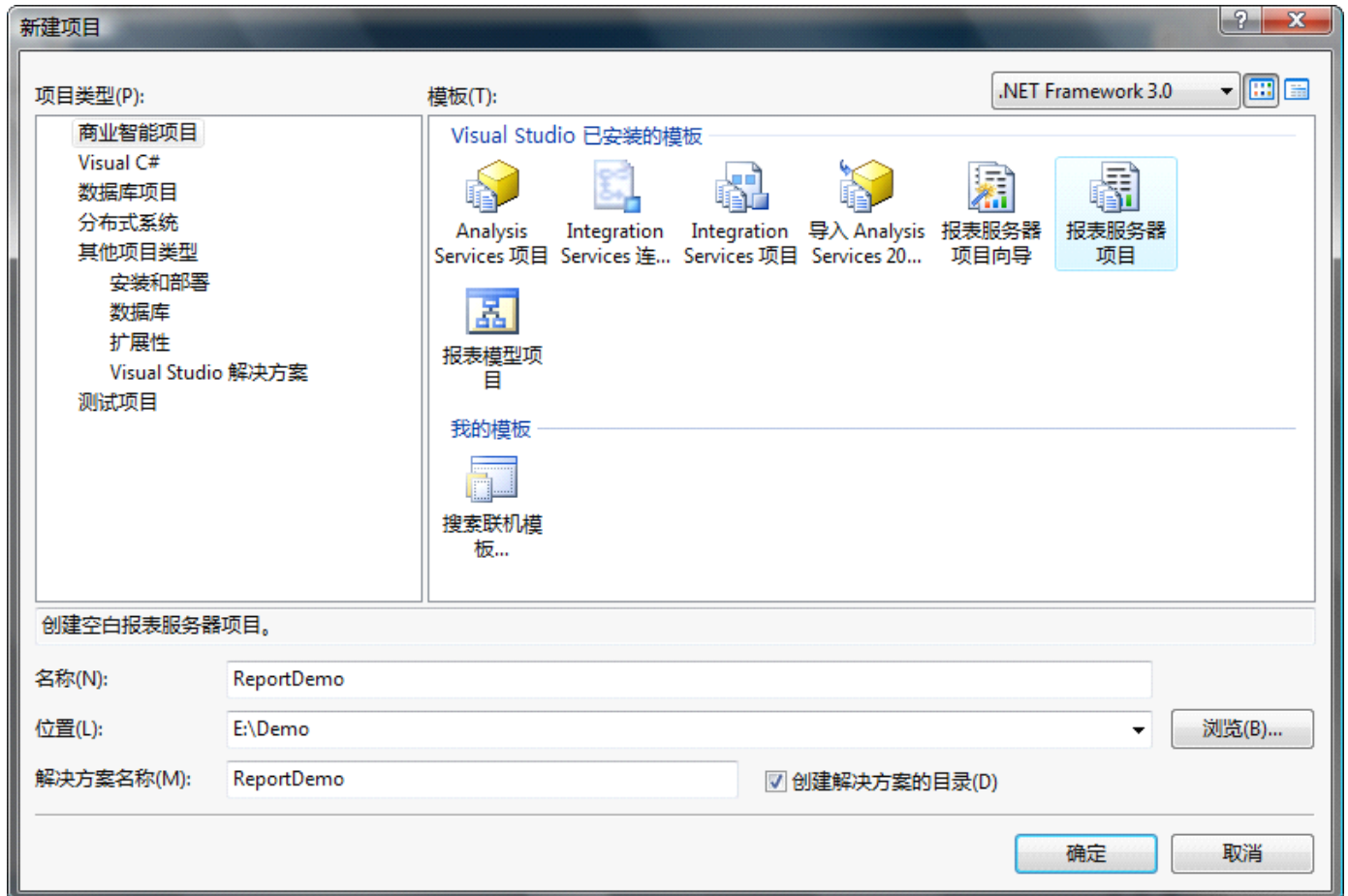
打开的开发界面如下：



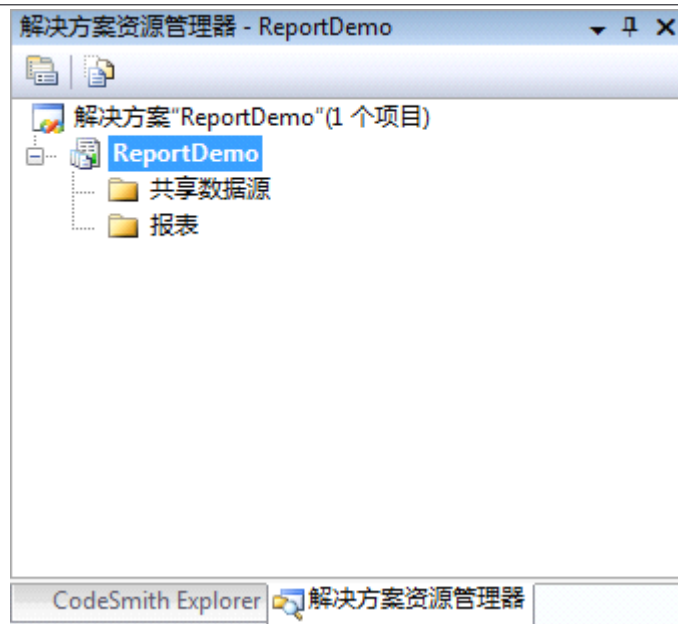
和一般的开发工具没什么区别，上面是菜单和工具栏，左边工具箱，右边是资源管理器，下面是状态及结果栏。大家在学习的过程中会慢慢的熟悉它。

下面让我们来看看如何新建一个报表项目。

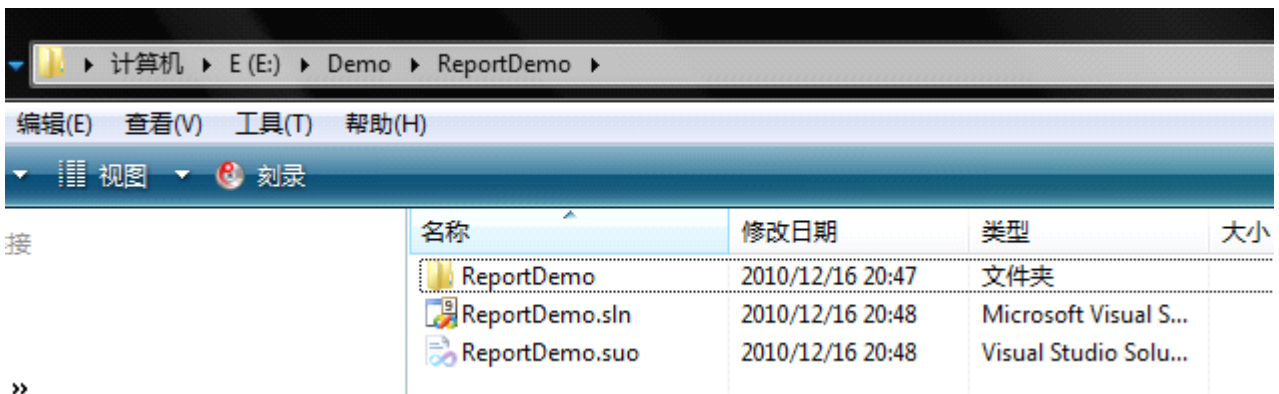
首先从“文件”菜单->“新建”->“新建项目” 出现新建项目界面，如下图：



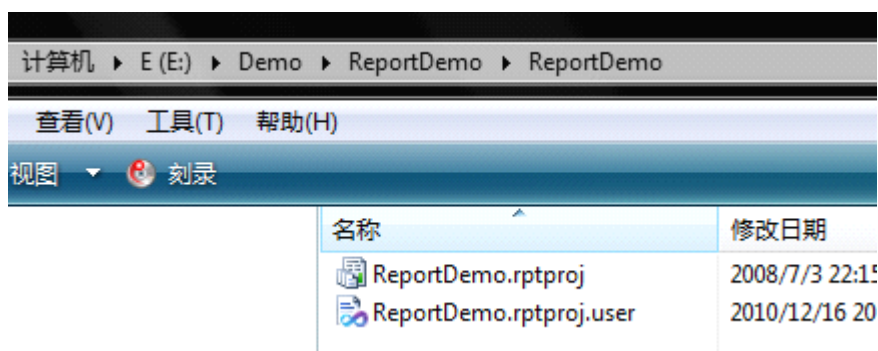
我们选择报表服务项目，然后填写项目名称和解决方案，选择项目的存放位置。点击“确定”后，我们可以在“资源管理器”窗口看到项目已经创建成功，如下图：



同时我们来看一下存放的目录：

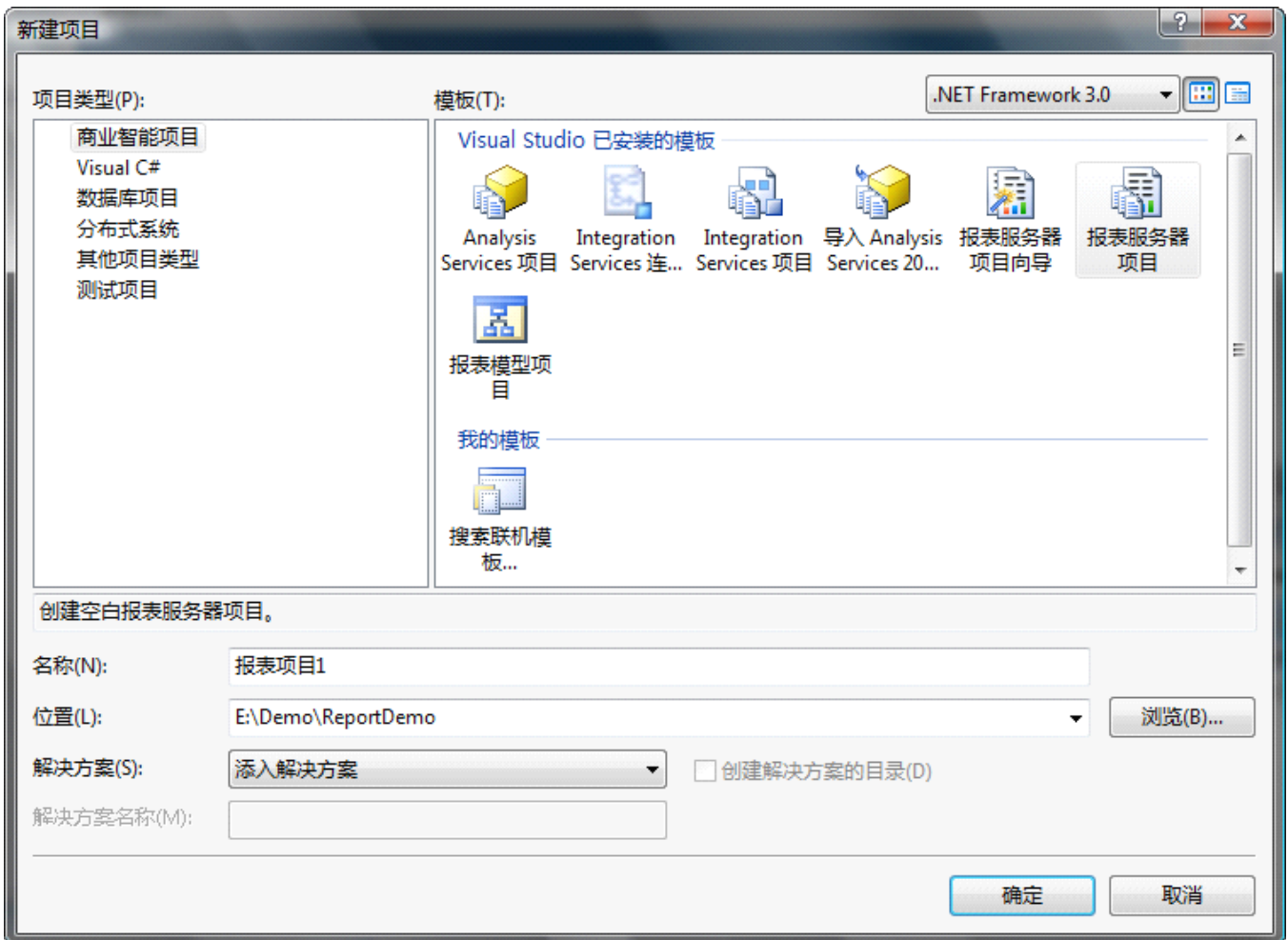


我们新建项目时，选择的存放位置为：E:\demo，并且勾选了“创建解决方案的目录”，所以第一层 ReportDemo 目录下存放着解决方案，第二层的 ReportDemo 才为报表项目的文件夹。如图：



在这里提醒大家的，为了方便管理，可以在同一个解决方案下添加多个报表项目，比如：为了报表分类管理，可以把员工的报表和公司的报表放在同一个解决方案下的不同报表项目中。

如果您已存在了解决方案，那么新建报表项目的时候，会出现如下图，在解决方案一栏，提示是“新建解决方案”还是“添入解决方案”。



新建数据源

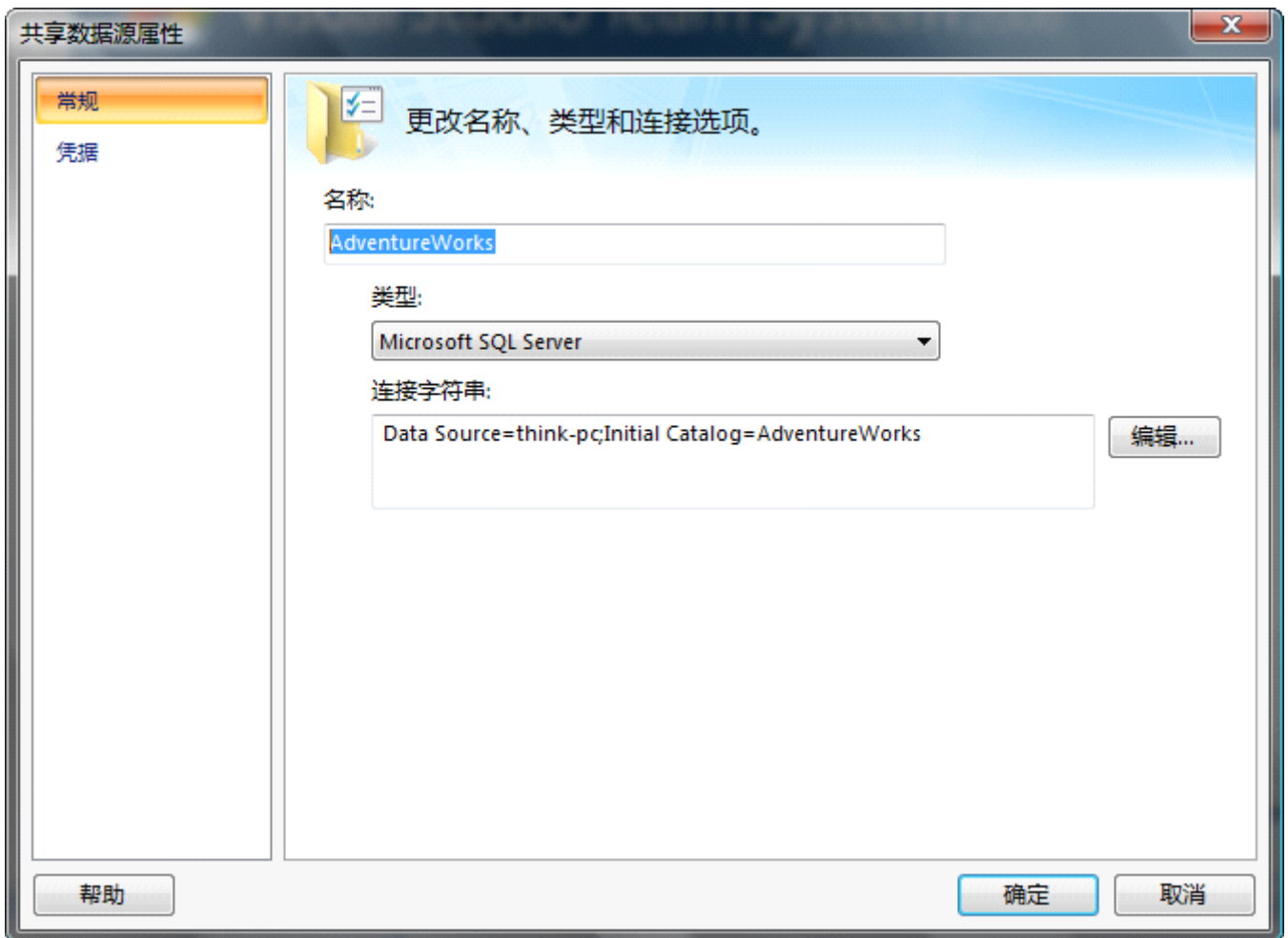
OK,项目的新建完成了，我们来看看如何新建数据源~~，这里以 AdventureWorks 示例数据库为例，后面的实例也依赖于该数据库。如果对 AdventureWorks 不了解的。可以参考如下地址中的数据字典。

[http://msdn.microsoft.com/zh-cn/library/ms124438\(v=SQL.100\).aspx](http://msdn.microsoft.com/zh-cn/library/ms124438(v=SQL.100).aspx)

我们右键“资源管理器”中的“共享数据源”，选择新建数据源。如图：



出现如下新建界面：



在“常规”选项卡中，填入数据源名称。选择类型为“Microsoft SQL Server”。

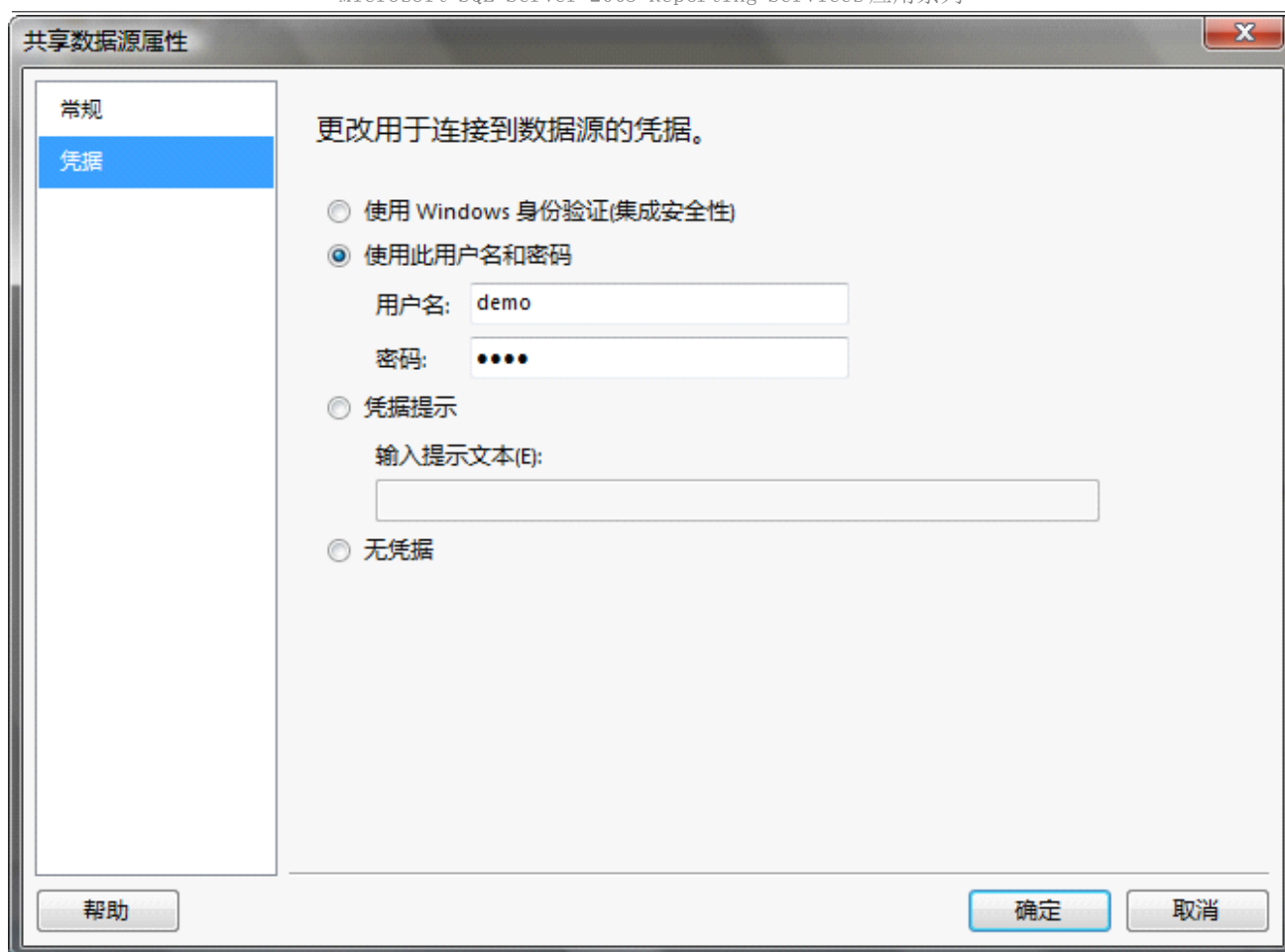
点击“编辑”按钮进入如下界面，编写连接字符串。



选择服务器名，登录方式及数据库，然后“测试连接”是否成功。我们这里以 demo 用户连接。

也可以在“高级”中设置其他属性，比如连接池数量，timeout 等。

“凭据”选项卡里的内容和上图中的登录方式是一个意思。都是指以什么样的身份连接到数据库。在连接属性中设置好了，也可以在“凭据”选项卡修改登录方式。凭据的设置界面如下图：



凭据可以设置为：windows 身份验证，指定用户等。如果设置凭据提示，那么运行报表时，将提示输入用户名及密码，显示的提示内容为这里设置的内容。

新建报表

报表项目建好了，共享数据源我们也建好了，那么下面让我们以一个实例来学习一下如何新建报表及使用基础控件。首先来看一下我们需要实现的报表：

该报表上包含“页眉，主体和页脚”，用到“文本框”，“线条”，“表格”以及页码的设置。该报表存在一个“CategoryName”参数。

当输入 CategoryName 参数，报表将显示该分类的子类。

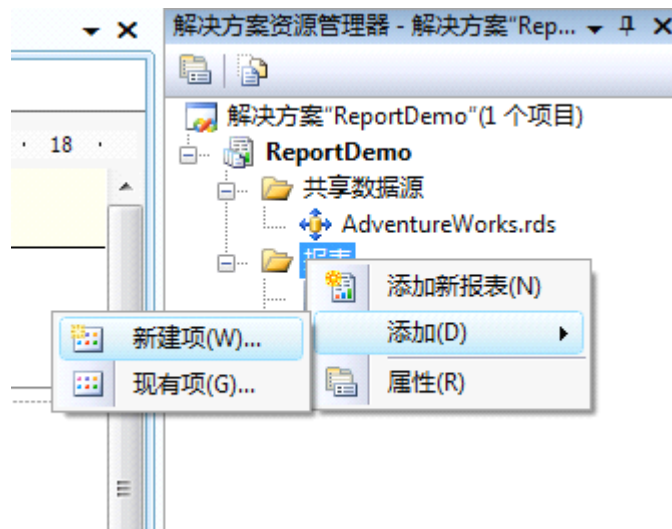
产品分类列表

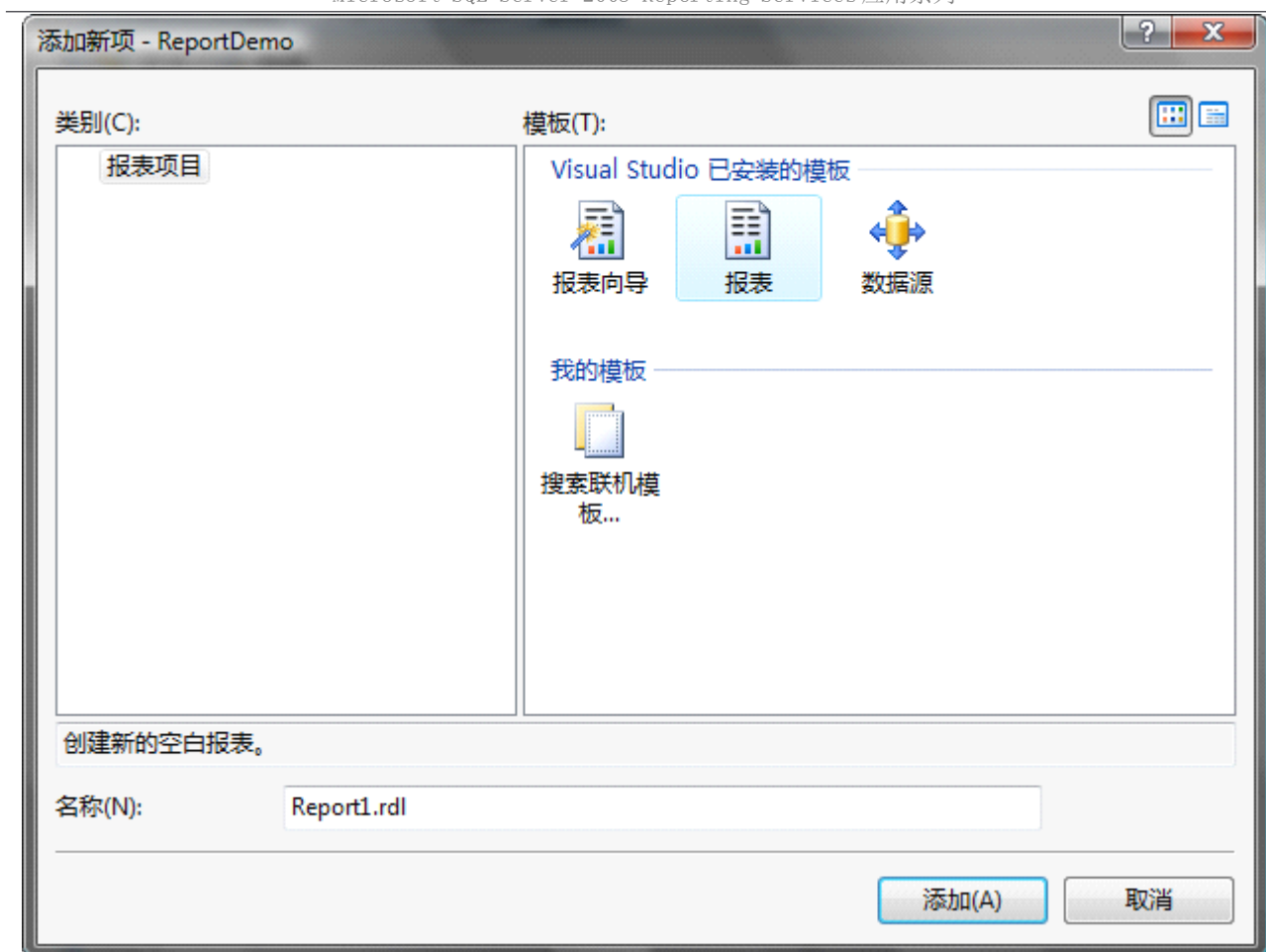
子类列表

Product Subcategory ID	Name	Modified Date
1	Mountain Bikes	1998/6/1 0:00:00
2	Road Bikes	1998/6/1 0:00:00
3	Touring Bikes	1998/6/1 0:00:00

1

看了效果图后，让我们自己动手完成吧！首先新建报表，右键报表文件夹，添加->新建项 如图：





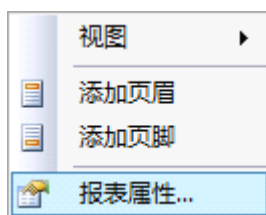
进入新建报表界面，在模板中我们选择报表，名称默认为 Report1.rdl。点击添加。我们可以看到在“资源管理器”下的文件夹下已经添加了“Report1.rdl”文件。当然也可以通过“报表向导”根据提示一步步来完成。不过如果要设置复杂的无规律的报表，向导将无法完成。

我们双击该文件，将在设计区出现空白的的设计界面。如图：

将项添加到报表：将某项从工具箱拖到设计图面，然后将数据集字段拖到该项。

我们先来设置报表的属性。先把我们想要的框架画出来。这样答应出来才能和我们设计的效果相同。

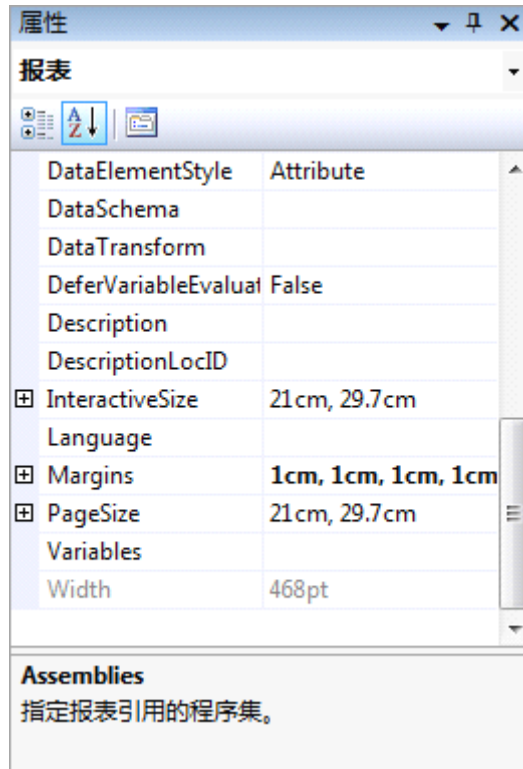
右键设计区域的空白处。将弹出如下图的菜单



选择报表属性，进入属性界面：

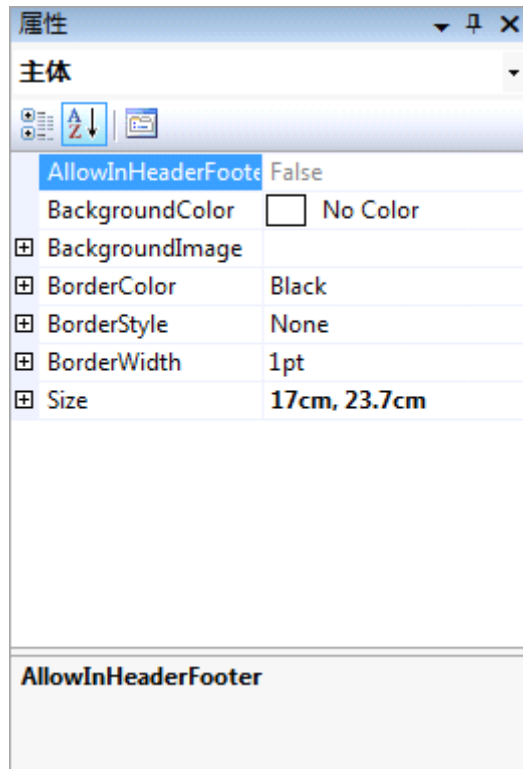


在此我们设置报表适应 A4纸，为了国家资源，我们把边距都设置成1cm. 点击“确定”完成设置。该操作也可以通过右边的属性栏来设置。如图：



那么如何添加页眉和页脚呢？其实很简单，通过同样方式，右键设计区域的空白处添加页眉和页脚。

下面我们通过属性窗口设置主体的宽和高。



本例中设置为宽：17CM,高：23.7CM。通过同样的方式设置页眉和页脚的高度为2CM。

注意：这里设置的总高度和总宽度，不要超出纸张的高度和宽度，不然打印的时候会多出空白页。如果存在空白页，可以设置报表主体的宽和高略微小于纸张的宽和高。

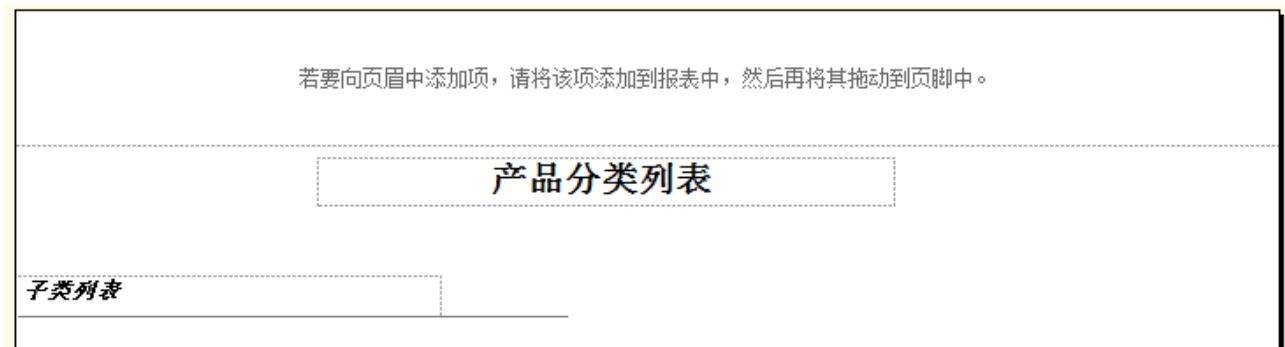
OK,前面这些准备工作也做的差不多了。我们应该进入主题了。

“文本框”使用

首先从“工具栏”拖入“文本框”控件，用来存放标题。调整文本框的长宽以及位置。在文本框内输入“**产品分类列表**”，字体要设置成中文字体，不然会乱码。

用同样方法在拖入一个“文本框”控件，调整文本框的长宽以及位置，在文本框内输入“**子类列表**”，同时拖入“线条”控件，放在该文本框下作为分割线。也可以通过设置文本框的下边框显示，来代替线条。

初步设置后如下图：

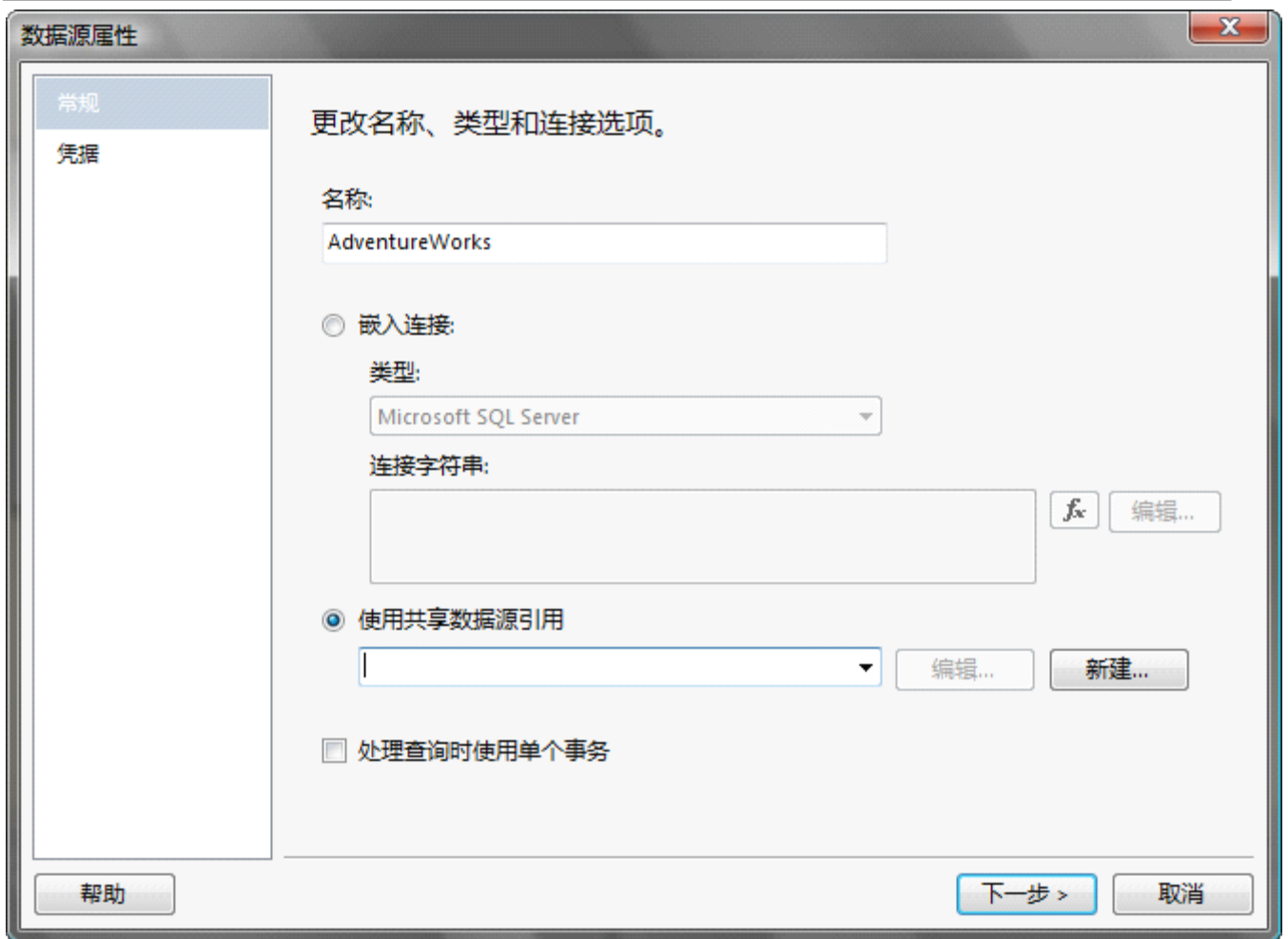


添加“Tablix”控件

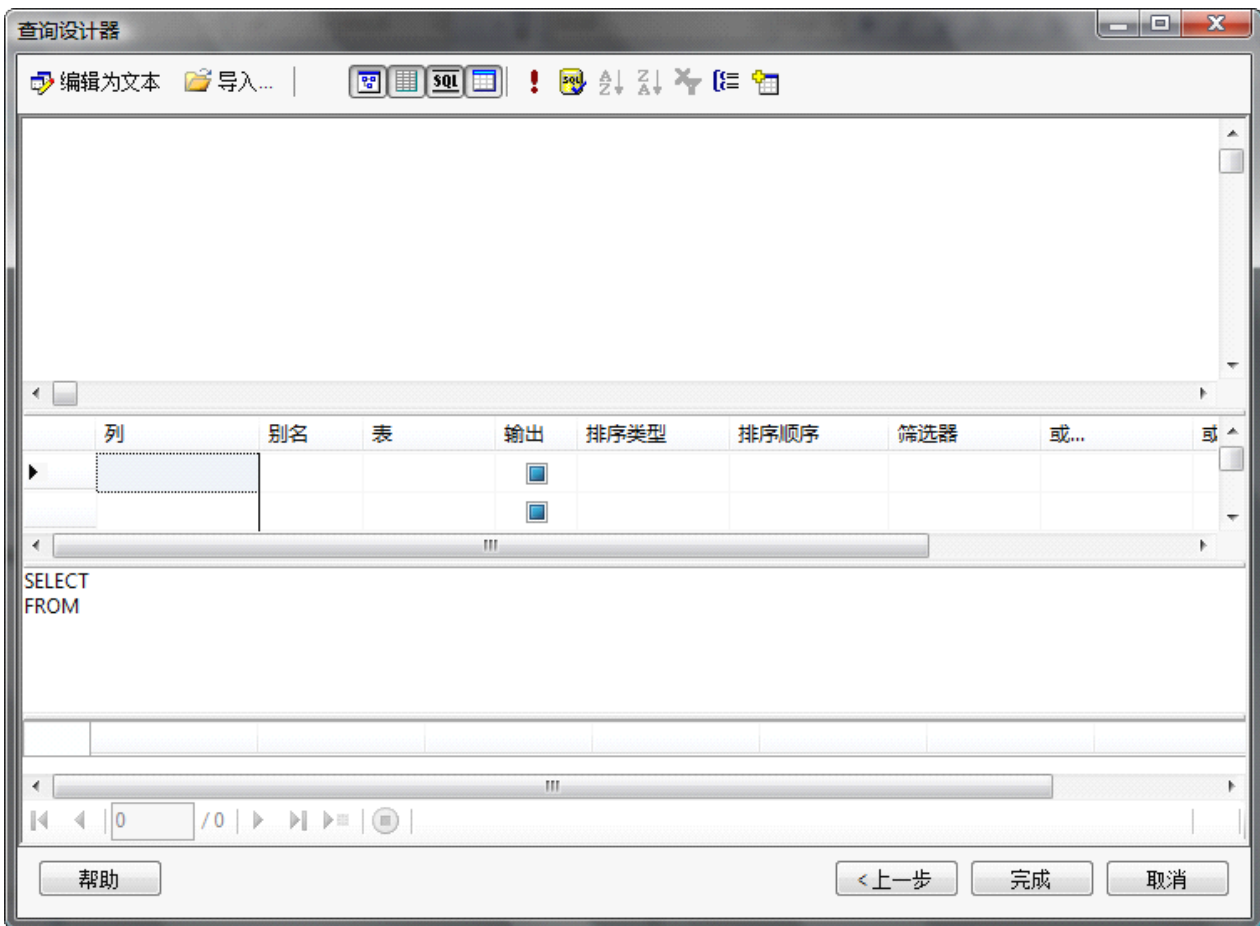
下面我们来添加“表”控件。从工具栏拖入表格时，因为表格需要绑定数据集，所以会提示设置报表的数据源属性。如下图：

在此我们使用“共享数据源”，在下拉列表中选择我们开始建立的“AdventureWorks”共享数据源。也可以通过“嵌入连接”来设置独立数据源。那么只有该报表可以使用。其他报表将不可以访问该数据源。设置方式和共享数据源的设置相同。

在此设置连接名称为：“AW”，名称可以任意设置。

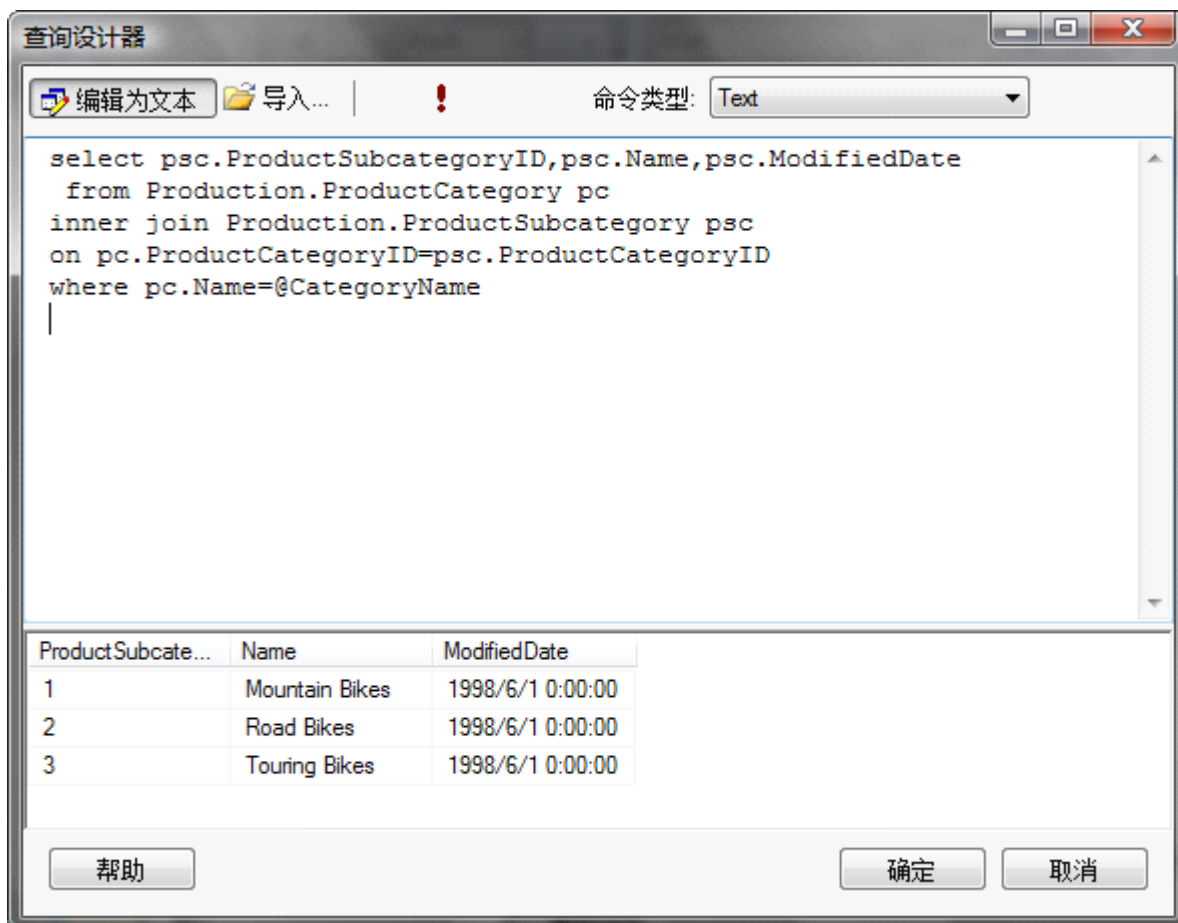


点击下一步将进入“查询设计器”，在此设置查询语句，如图。

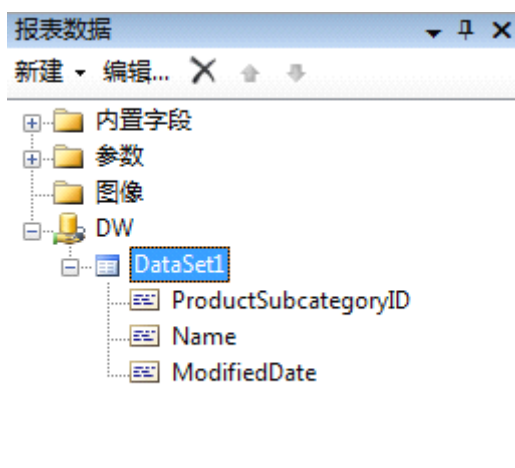


通过上面的功能按钮及设计窗口，可以很容易的设计查询语句，我们也可以在数据库的“查询器”中设计好语句粘帖到这里。不过如果直接粘帖到查询设计器中，我们的 SQL 语句会被重新格式化。

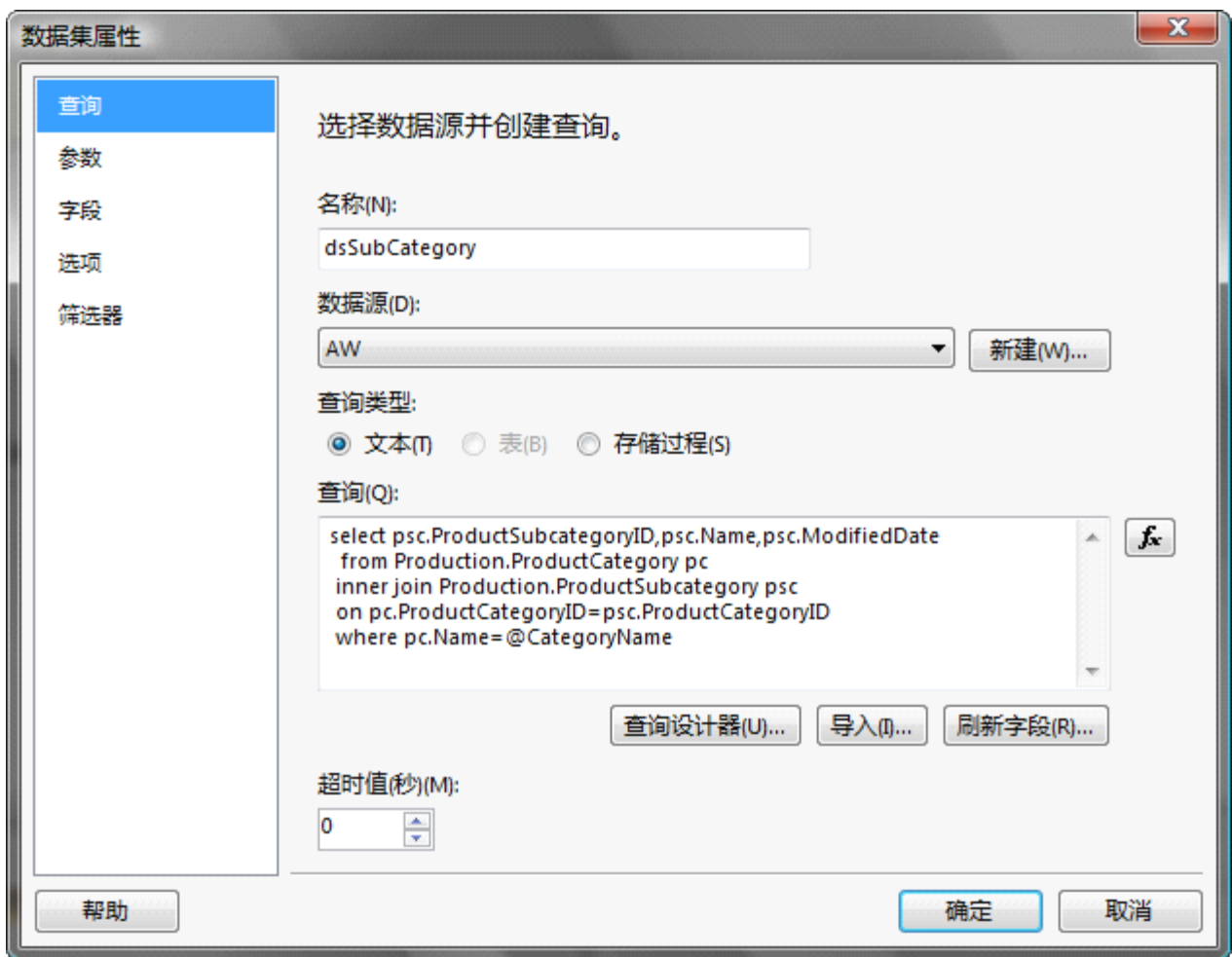
如果不想被改变编写格式，我们可以点击“编辑为文本”，然后再粘帖。将保持我们原有的格式。如图：



点击“执行”，我们看到的结果和在数据库的查询器中的结果相同。点击“确定”后，我们可以在“报表数据”管理器中看到新建了数据集 Dataset1。



我们可以双击数据集的名称来修改它的属性。我们重新命名数据集名称为：dsSubCategory，SQL 语句中的查询字段，参数 会自动添加到“字段”和“参数”选项卡。这里也可以设置其他的一下属性，比如排序规则，对查询结果进行筛选等等。大家可以试着用用。如图：



修改 DataSet 的名称，我们同时需要修改表格绑定的 DataSet 名称。同时大家应该可以看到“查询类型”一项，这里可以选择文本还是存储过程，也就是说您可以把 SQL 查询语句写成存储过程，再在报表中调用。

SQL 脚本为：

```
select  psc.ProductSubcategoryID,psc.Name,psc.ModifiedDate

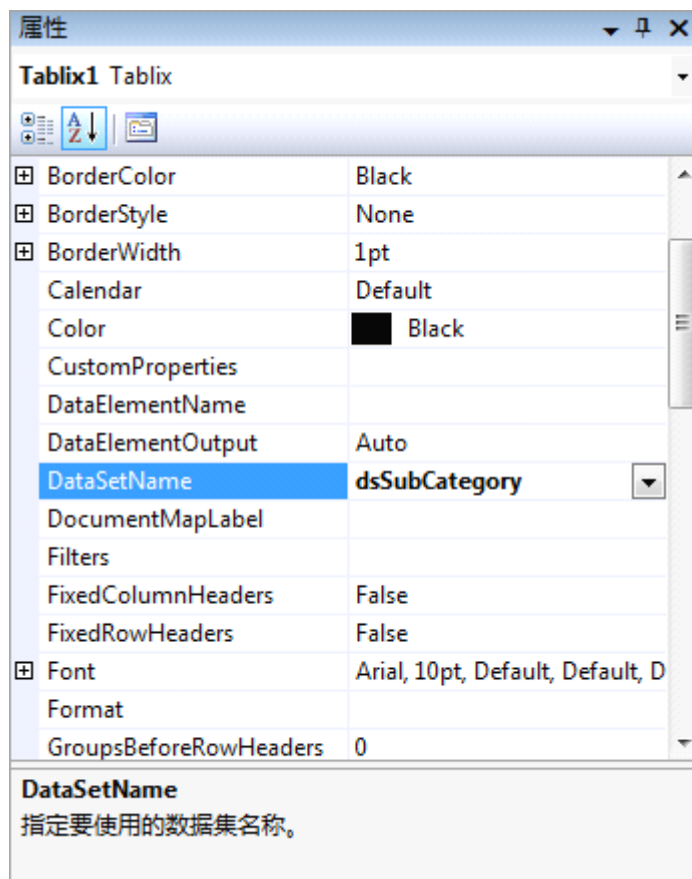
        from Production.ProductCategory pc

inner join Production.ProductSubcategory psc

        on pc.ProductCategoryID=psc.ProductCategoryID

        where pc.Name=@CategoryName
```

下图为设置 Tablix 的数据集的属性窗口：



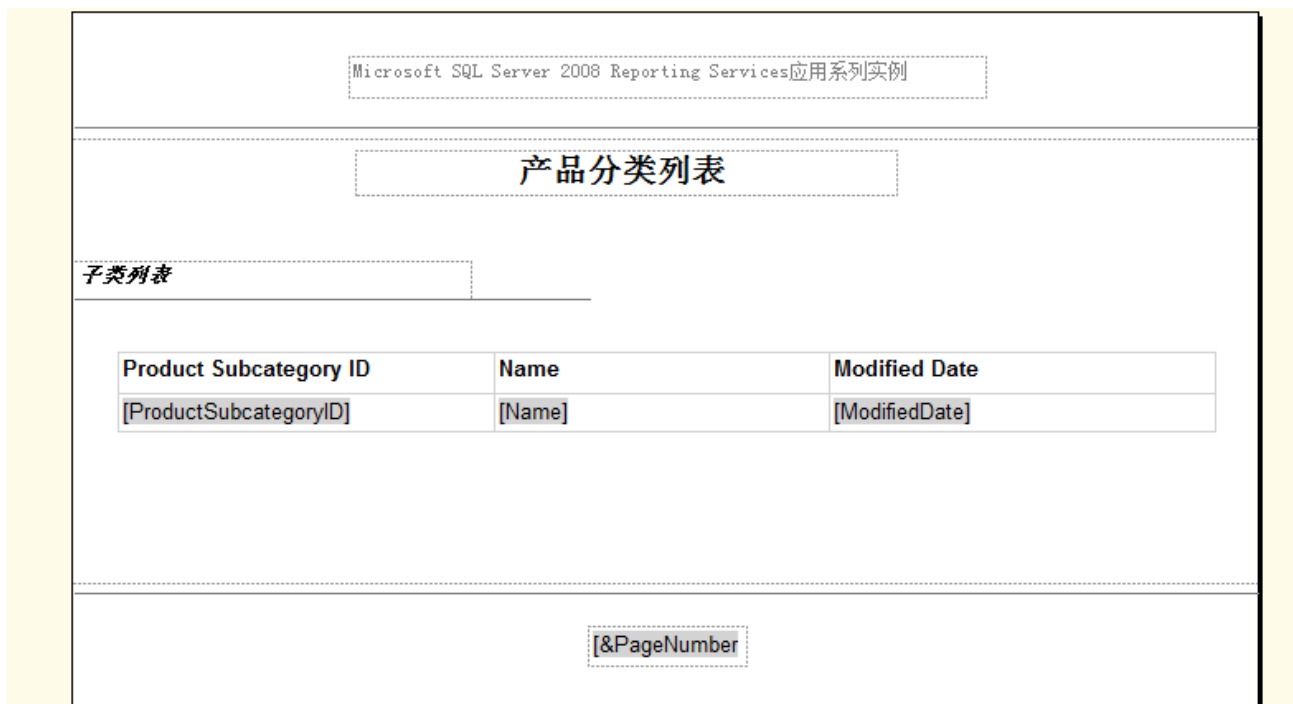
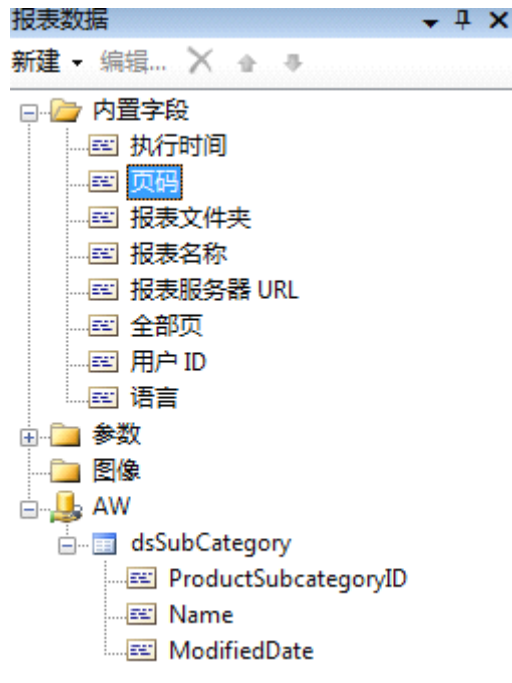
数据集的属性设置好了后，我们来把数据集的数据放到表格的具体位置。

具体方法为：拖动数据集的字段到表格的列中即可。相应的列标题也会填上字段的名称，您可以自己修改成想要的名称。如果不通过拖动来添加，也可以通过表格中的文本框的表达式来设置。方法多多了～～不再阐述。设计好的界面如图：

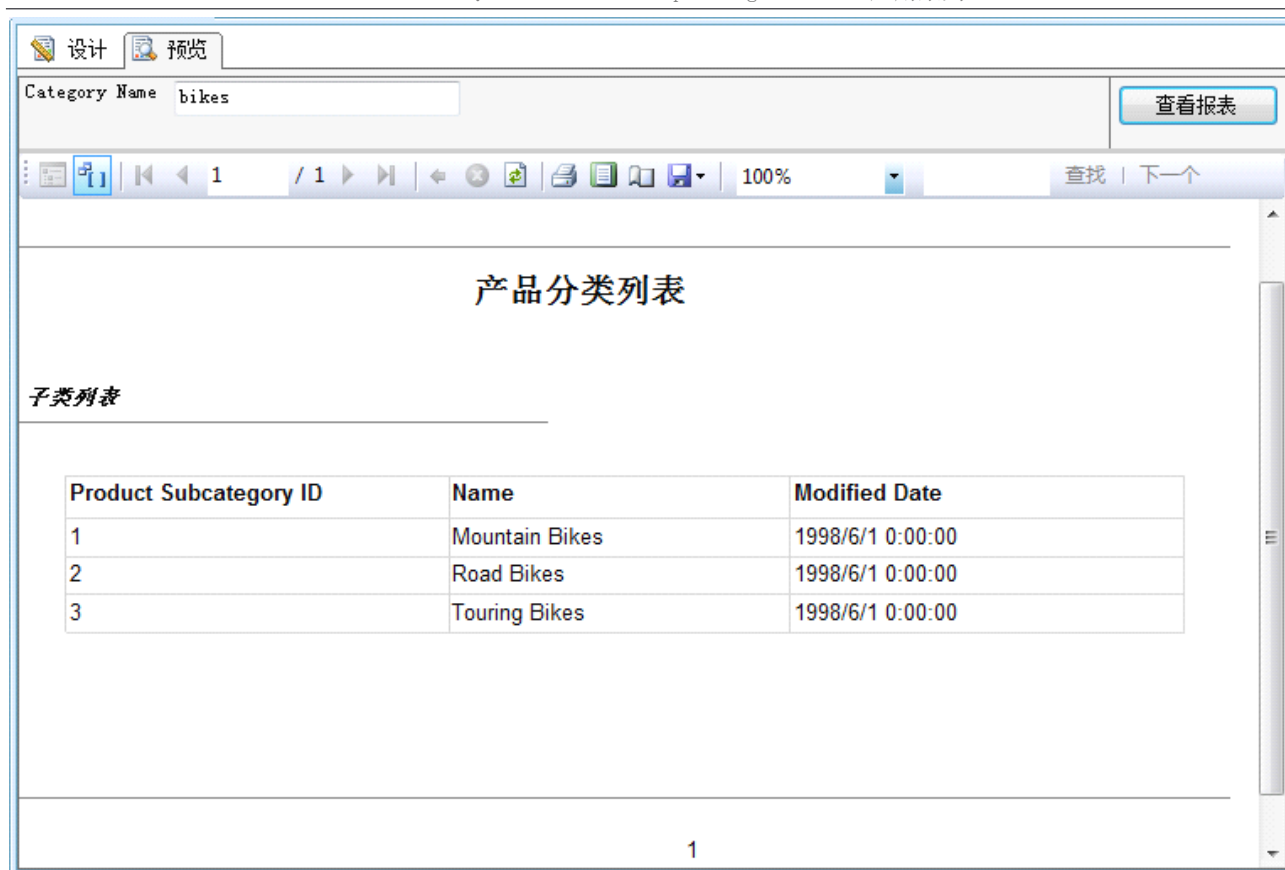


报表的主体已经设计好了，那么页眉和页脚设计也很简单，“页眉”我们按照前面的方法，拖入文本框和线条做相应的调整就 OK 了。

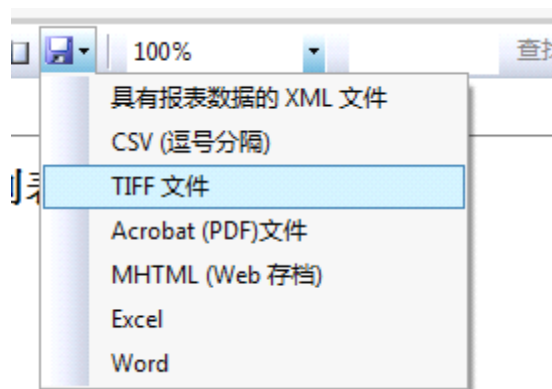
页脚的页码，我们可以从“内置字段”拖入页码也就 OK。如图：



终于设计完了，让我们来看看效果。点击“预览”。输入参数值“bikes”，效果如图：



我们可以通过查看器上的导出功能导出文件。支持的格式如图：

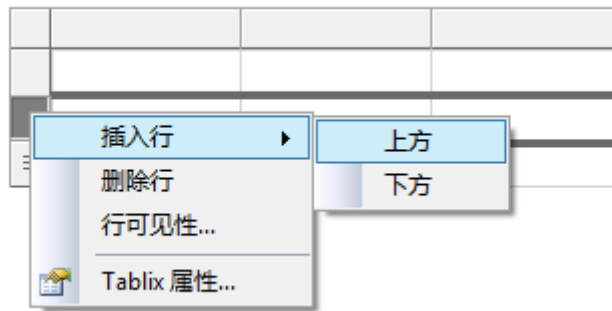


Tablix 控件介绍

经过前面的基础学习，我们已对 Tablix 有了初步的了解，下面我们来具体的学习一下 Tablix 的分组汇总等一些复杂的功能。先介绍一下 Tablix 主要组成部分。Tablix 包含：页眉，数据行，数据列，页尾，行组，列组。当我们从工具栏拖入设计区时，应该如下图：

	页眉	
	数据	

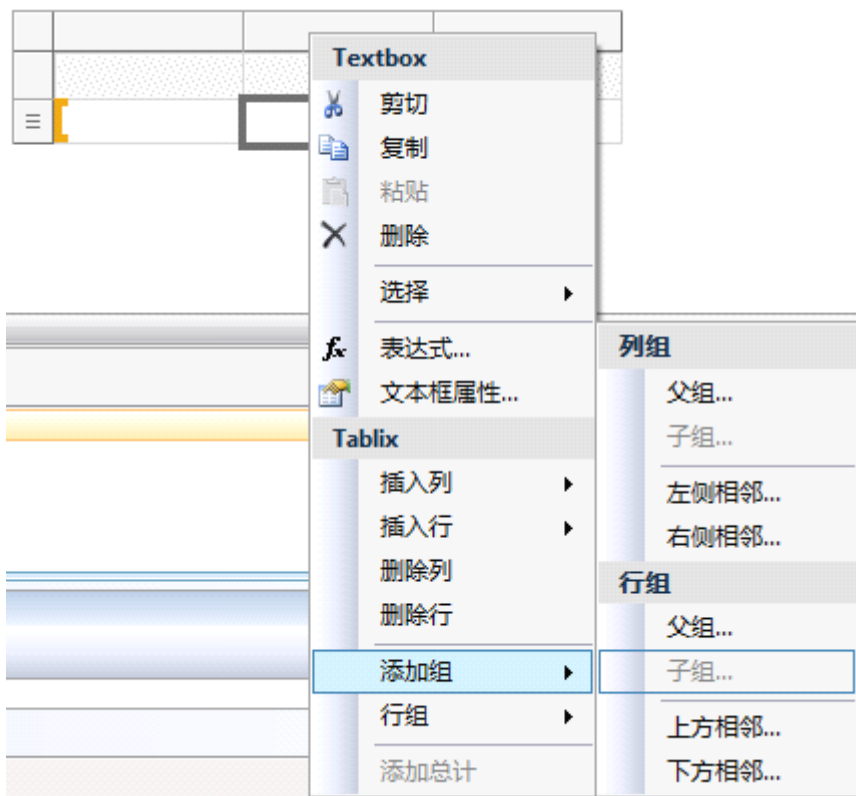
可以看到页眉行及数据行，我们可以通过右键页眉行来添加多条页眉。如图：



我们可以通过如上方法，实现如下效果，把上面的一行页眉“合并格式”，用来填入 Title，这种方式经常使用。

Road Bikes 销售记录		
Product Name	Order Qty	Line Total
Mountain-100 Black, 44	2	4049.988000
Mountain-100 Silver, 48	3	6119.982000
Mountain-100 Black, 38	2	4049.988000
Mountain-100 Black, 48	2	4049.988000
Mountain-100 Silver, 42	2	4079.988000
Mountain-100 Black, 38	1	2024.994000

也可以通同样操作来添加行组及列组。不过是右键“单元格”；如下图：上方相邻，下方相邻指组放在数据行的上方还是下方。



我们来看一下“上方相邻的行组”的效果。我们这里设置按 Order Date 分组，大家可以看到下图中，会先显示汇总的信息然后再显示明细数据，这就是非嵌入分组。如果我们插入的是“父级”分组，那么将形成嵌入分组。

Road Bikes 销售记录			
Order Date	Product Name	Order Qty	Line Total
2002/1/1 0:00:00		171	348928.970000
2002/1/3 0:00:00		3	10149.970000
2002/1/4 0:00:00		1	3399.990000
2002/1/1 0:00:00	Mountain-100 Silver, 48	1	2039.994000
2002/1/1 0:00:00	Mountain-100 Black, 42	2	4049.988000
2002/1/1 0:00:00	Mountain-100 Black, 44	2	4049.988000

再来看看插入列组的效果：如下的日期列为动态生成，这里是统计出每种产品每天卖出的数量。

设计界面如：

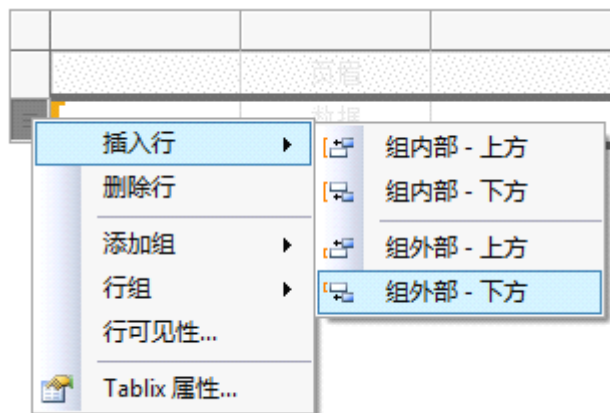
Product Name	Unit Price	[OrderDate]
[ProductName]	[UnitPrice]	[Sum(OrderQty

效果:

Product Name	Unit Price	2002/2/2 0:00:00	2002/2/3 0:00:00
Road-150 Red, 52	3578.2700	1	
Road-150 Red, 52	3578.2700	1	
Road-150 Red, 62	3578.2700	1	
Road-650 Black, 48	699.0982	1	
Road-150 Red, 48	3578.2700	1	
Road-150 Red, 52	3578.2700	1	
Road-150 Red, 48	3578.2700		1
Road-150 Red, 52	3578.2700		1
Road-150 Red, 52	3578.2700		1
Road-150 Red, 48	3578.2700		1
Road-150 Red, 52	3578.2700		1
Road-150 Red, 48	3578.2700		1

插入页脚可以通过右键“数据行”，在菜单中选择“组外部-下方”如图:

当对所有数据汇总或运算时，可以通过该方法插入页脚。



Tablix 实例（一）

上面的基本操作学会了，可以达到的效果也看了，那么让我们来小试牛刀~~，学习一下嵌入式分组。

我们将要实现的报表如下图，下面的实例我们将用到 Tablix 行组及实现由用户交互控制数据的显示。

Order Date	Sales Order Number	Product Name	Order Qty	Line Total
2002/1/1 0:00:00			171	348928.970000
2002/1/3 0:00:00			3	10149.970000
	SO45087		1	3374.990000
	SO45088		1	3399.990000
	SO45089		1	3374.990000
			174	359078.940000

上图中可见首先对详细数据按订单日期分组并汇总出“数量”，“总金额”，再按订单分组详细数据并汇总出每个订单的“数量”，“总金额”。当用户点击前面的+ 图标，将逐层展开明细。

然后来看看设计界面，从设计界面可看到这里包含两个嵌套分组。

Order Date	Sales Order Number	Product Name	Order Qty	Line Total
[OrderDate]			[Sum(OrderQty)]	[Sum(LineTotal)]
	[SalesOrderNumber]		[Sum(OrderQty)]	[Sum(LineTotal)]
		[ProductName]	[OrderQty]	[LineTotal]
			[Sum(OrderQty)]	[Sum(LineTotal)]

通过下图的“行组”窗口可以看到各个分组之间的关系，OrderDate 分组包含 OrderID 分组，OrderID 包含明细数据。



下面我们来一步一步实现它。首先按照前面学习的方式，建立数据集，拖入 Tablix 控件并绑定数据集。

数据集 SQL 脚本如下：（上面的报表也可以参考该处的脚本）

```
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

INNER JOIN Production.ProductSubcategory PSC

ON P.ProductSubcategoryID=PSC.ProductSubcategoryID

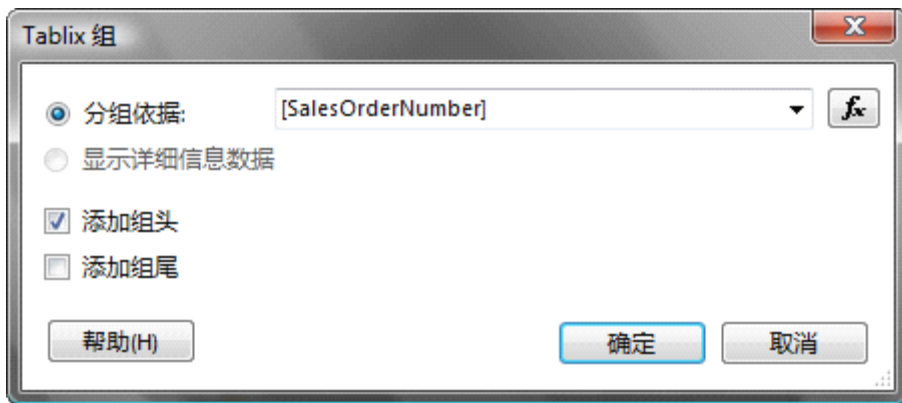
where PSC.Name='Road Bikes'

AND OrderDate BETWEEN @StartDate AND @EndDate
```

初步工作完成后，设计界面应该如下图：有三列 Product Name, Order Qty, Line Total 。我们点击数据行来添加“父组”。



点击添加“父组”后，弹出如下窗口：设置分组依据，勾选添加组头。



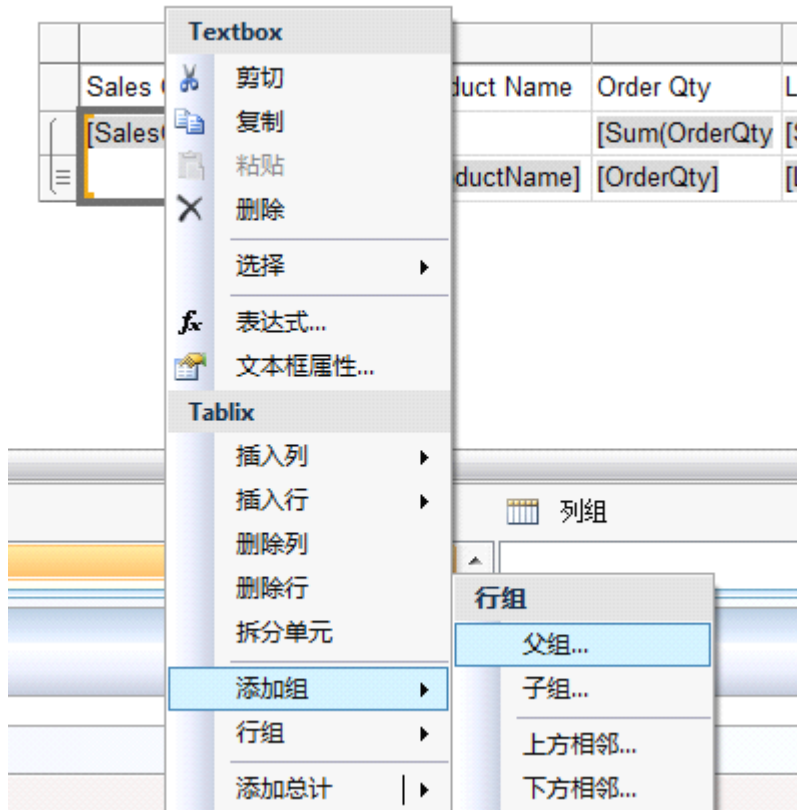
点击“确定”后，应该是如下界面：

Sales Order Number	Product Name	Order Qty	Line Total
[SalesOrderNumber]			
	[ProductName]	[OrderQty]	[LineTotal]

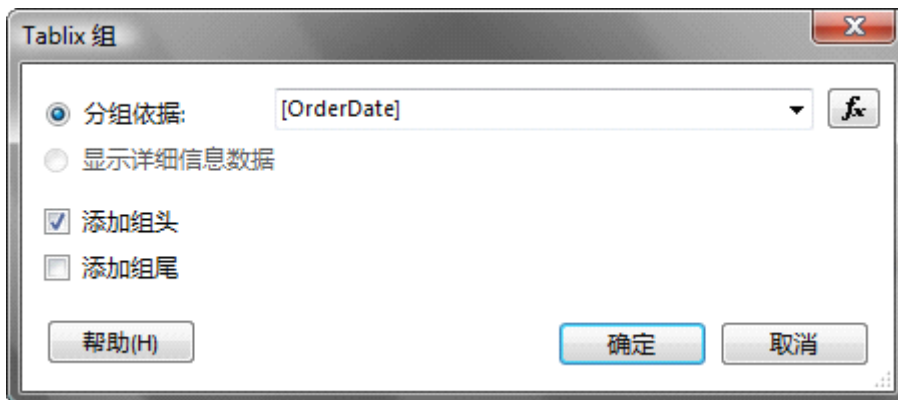
我们在 Order Qty , Line Total 做汇总操作。

Sales Order Number	Product Name	Order Qty	Line Total
[SalesOrderNumber]		[Sum(OrderQty]	[Sum(LineTotal]
	[ProductName]	[OrderQty]	[LineTotal]

OK， 第一层的分组已经设置好了，我们在做按 Order Date 分组。点击“SalesOrderNumber”分组框。弹出如下界面：



选择“父组”进入设置界面：设置分组依据为：[OrderDate],添加组头。



设置好了分组后，同样做汇总操作，我们在这里在添加一个页脚来汇总所有的数量级金额。设置了两层嵌套分组的 Tablix,如下图：

Order Date	Sales Order Number	Product Name	Order Qty	Line Total
[OrderDate]			Date Total: [Sum(OrderQty)]	[Sum(LineTotal)]
	[SalesOrderNumber]		Order Total: [Sum(OrderQty)]	[Sum(LineTotal)]
		[ProductName]	[OrderQty]	[LineTotal]
			Total: [Sum(OrderQty)]	[Sum(LineTotal)]

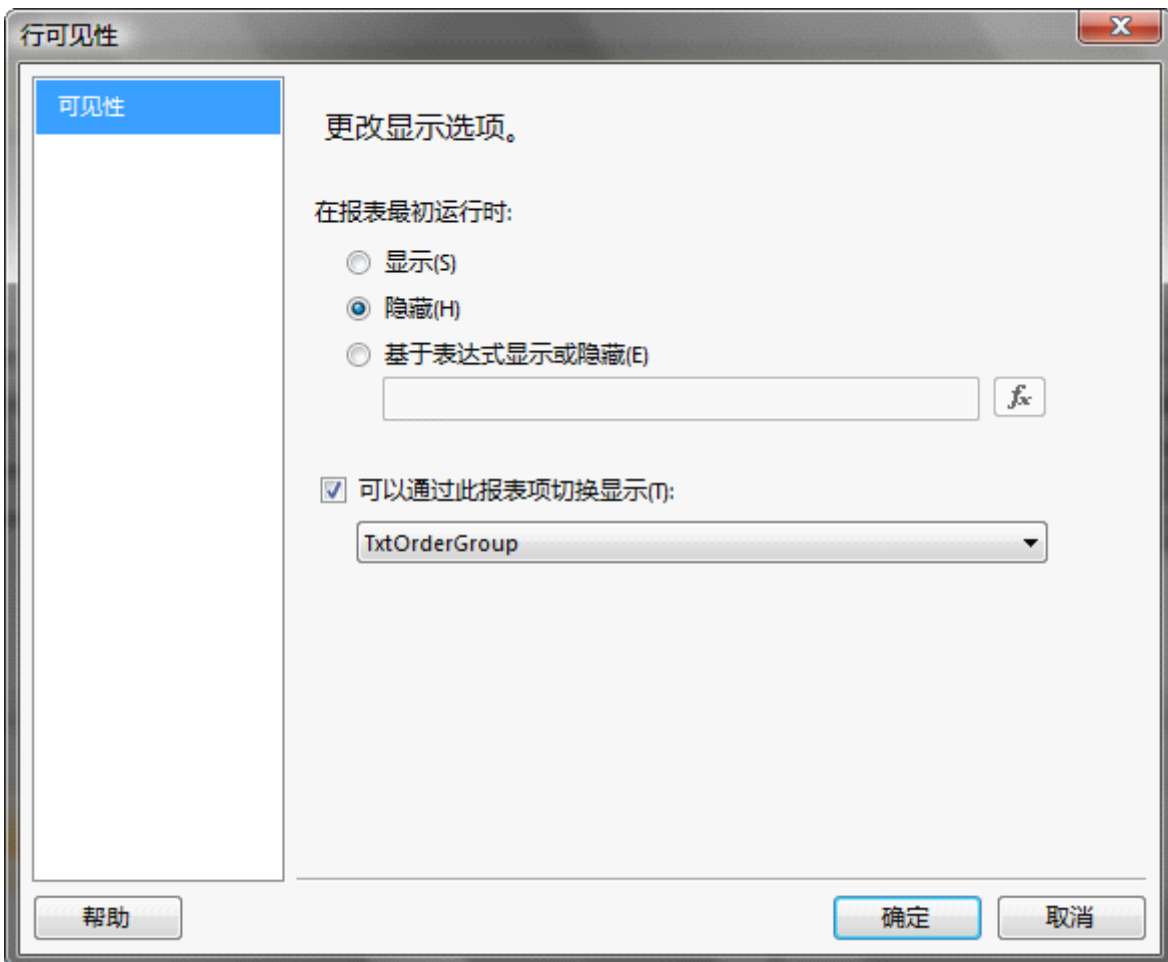
下面我们设置每层的显示依赖于父组层。首先设置详细数据依赖于 Order Group,右键数据行，在菜单中点

击“行可见性”，如图：

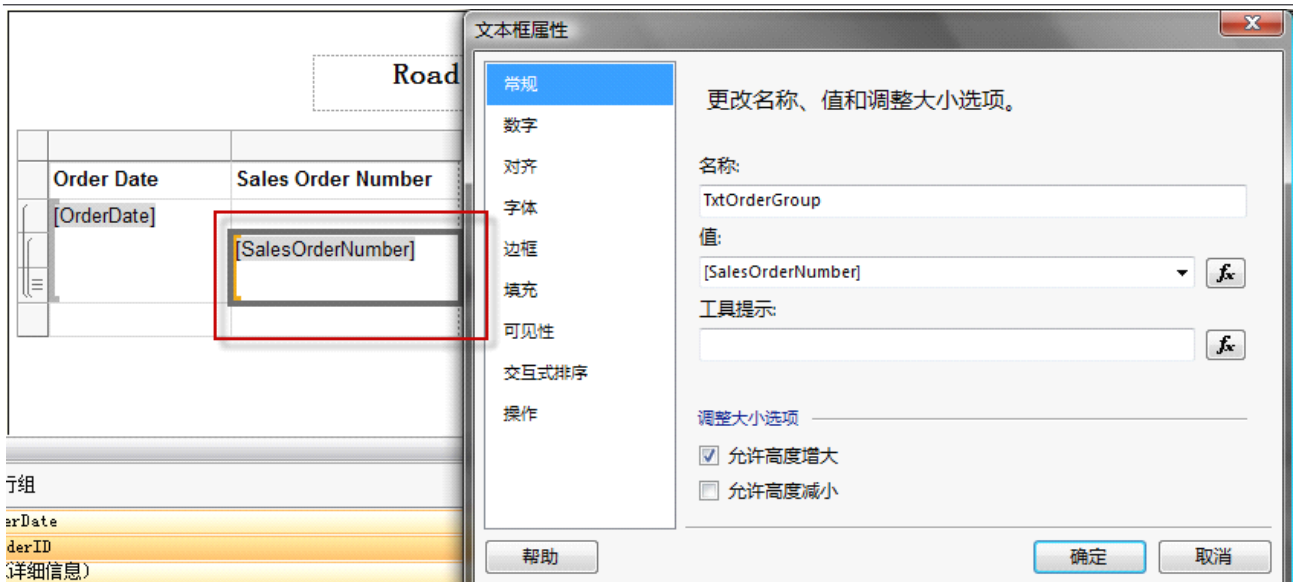
Order Date	Sales Order Number	Product Name	Order Qty	Line Total
[OrderDate]			Date Total: [Sum(OrderQty)]	[Sum(LineTotal)]
	[SalesOrderNumber]		Order Total: [Sum(OrderQty)]	[Sum(LineTotal)]
		[ProductName]	[OrderQty]	[LineTotal]
			Total: [Sum(OrderQty)]	[Sum(LineTotal)]

右键菜单选项：
 插入行
 删除行
 行可见性...
 Tablix 属性...

进入设置界面：

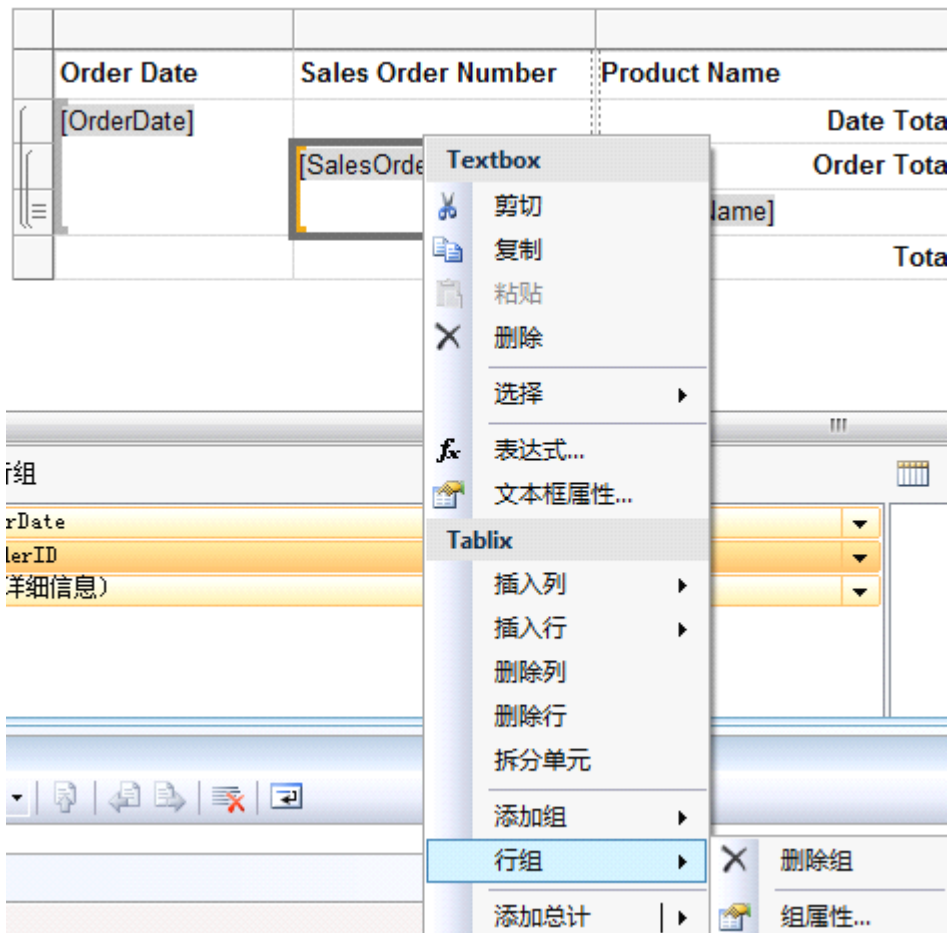


我们设置数据行初始运行时为“隐藏”。勾选“可以通过此报表项切换显示”。选择报表项为：TxtOrderGroup，该项为按订单的分组项，这里名称可以自行修改：修改方式如下图：



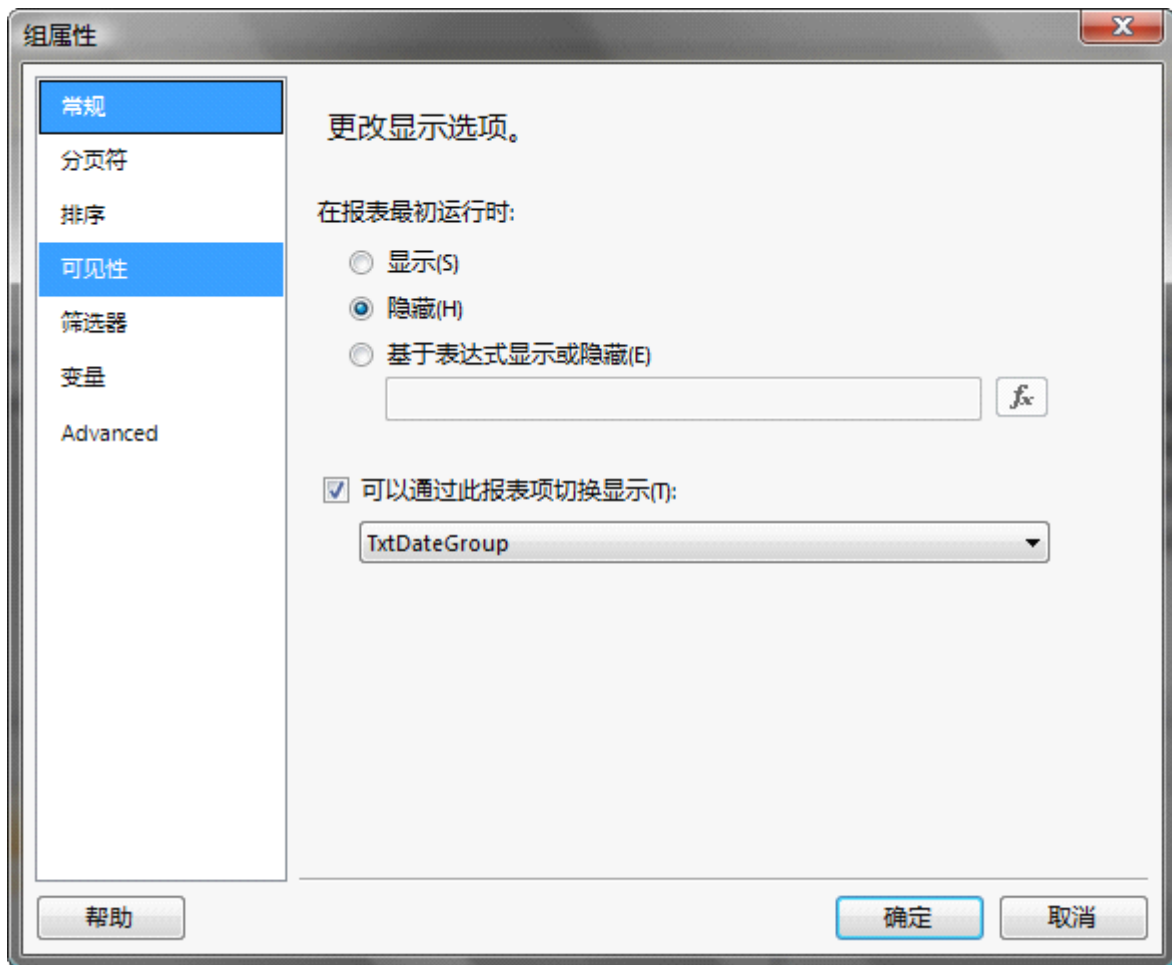
点击分组项,文本框属性, 设置名称即可。 OrderDate 分组项也可以同样设置。

第一层的控制设置好后, 再设置 Order Group 的显示依赖于 Date Group。如下图:



这地方要把控制设置在分组项上, 不能单纯设置在组头上。不然达不到效果。所以这里应该选择“组属性”

进入设置：



和上面的设置相同，设置隐藏，通过报表项切换。OK。设置完了，预览看看效果如何～～

设计 预览

Start Date 2002-1-1 End Date 2002-1-3 查看报表

1 / 2? 100% 查找 | 下一个

Road Bikes 销售记录

Order Date	Sales Order Number	Product Name	Order Qty	Line Total
2002/1/1 0:00:00		Date Total:	470	350178.840440
2002/1/2 0:00:00		Date Total:	4	14313.080000
		Order Total:	1	3578.270000
	SO45083	Order Total:	1	3578.270000
	SO45084	Order Total:	1	3578.270000
	SO45085	Order Total:	1	3578.270000
2002/1/3 0:00:00	SO45086	Order Total:	1	3578.270000
		Date Total:	5	17891.350000
		Total:	479	382383.270440

和我们的预期效果完全一样 o(∩_∩)o...

休息一下 o(∩_∩)o...

Tablix 实例（二）

下面给大家带来一个综合性的实例应用。该实例包含“静态列，行组，列组的结合使用。

该报表实现的功能是当输入开始时间，结束时间，将汇总出该区间内，每个客户每年的订单金额，及每月的金额，及详细的订单金额。

看看效果：

		⊕ 2002
Customer ID	Sales Order Number	Year Total
⊕ 2		9216.3596
⊕ 3		71112.1316
⊕ 4		238031.9559
⊕ 5		20720.6175
⊕ 7		6019.0371
⊕ 10		45089.9298
⊕ 12		79060.5774
⊕ 16		68272.1366
⊕ 17		10766.8293
⊕ 18		58420.5745

上图中的报表是根据输入的时间区间生成，可以很容易的知道每个客户每一年的订单金额。我们可以点击年份前的+，展开查看每个月的金额。如图：

		⊖ 2002				
Customer ID	Sales Order Number	Year Total	7	8	9	10
⊕ 2		9216.3596		9216.3596		
⊕ 3		71112.1316			71112.1316	
⊕ 4		238031.9559	118394.1258			119637.8301
⊕ 5		20720.6175			20720.6175	
⊕ 7		6019.0371	3569.4338			2449.6033
⊕ 10		45089.9298			45089.9298	
⊕ 12		79060.5774		79060.5774		

还可以点击 Customer ID 前的+ 展开该客户在该段时间内的每条订单的具体金额。如图：

			2002			
Customer ID	Sales Order Number	Year Total	7	8	9	10
2		9216.3596		9216.3596		
	SO46976			9216.3596		
3		71112.1316			71112.1316	
4		238031.9559	118394.1258			119637.8301
5		20720.6175			20720.6175	
7		6019.0371	3569.4338			2449.6033
10		45089.9298			45089.9298	
12		79060.5774		79060.5774		
16		68272.1366			68272.1366	
17		10766.8293	8789.9444			1976.8849
18		58420.5745		58420.5745		
20		9047.7586			9047.7586	
21		68309.5811			68309.5811	
22		2889.0273		2889.0273		

看了上面的效果以后，是不是有点跃跃欲试了，但是又担心这么复杂的报表是否做得出呀~~

如果前面介绍的知识您都能够掌握了，这个报表对您来说，应该是相当的简单~~

我们还是一起来看一下设计界面：

Customer ID	Sales Order Number	Year Total	[OMonth]
[CustomerID]		[Sum(SubTotal)]	[Sum(SubTotal)]
	[SalesOrderNumber]		[SubTotal]

通过上图可以看到我们设置了行组（Customer ID），列组(Omonth,Oyear)，及静态列(Sales Order Number)。

下面我们就一步一步实现它~~。

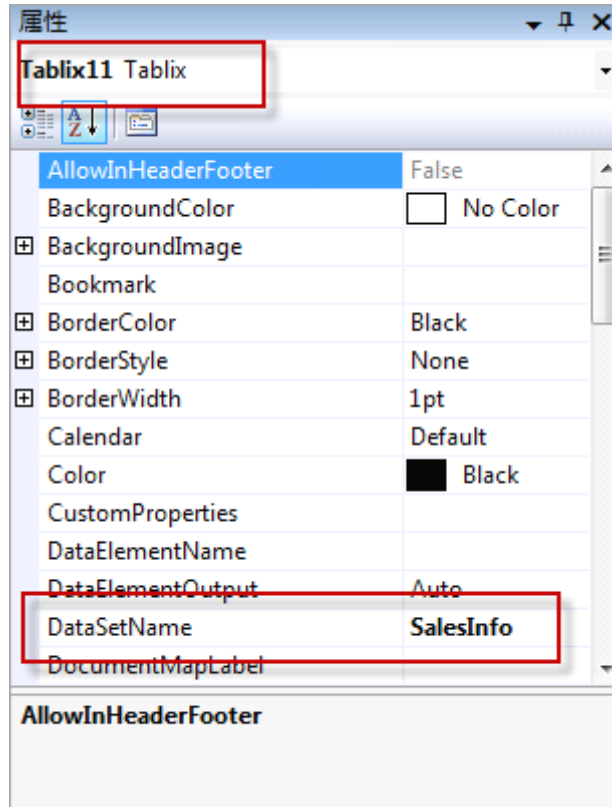
首先根据前面的方法新建报表，这不用说了，然后新建数据集~~数据集的脚本如下：

```
SELECT year(OrderDate) as Oyear,MONTH(OrderDate) as OMonth,SalesOrderNumber,
CustomerID,SubTotal
FROM Sales.SalesOrderHeader O
```


where OrderDate BETWEEN @StartDate AND @EndDate

以上步骤做了好以后，下一步当然是从工具栏拖入“Tablix”控件。把刚建好的数据集绑定到 Tablix。

从属性窗口选择数据源即可。如图：



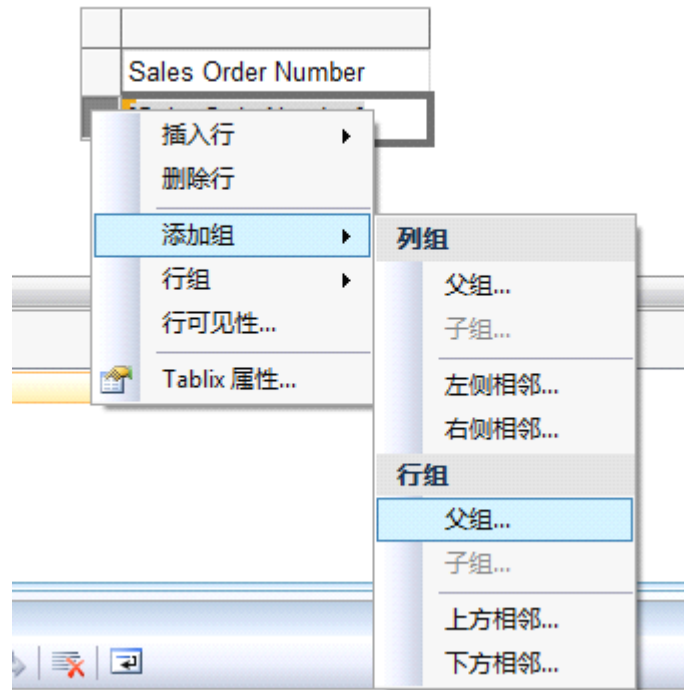
如果不熟悉的，可以回到前面温习一下~~，磨刀不误砍柴工，一定要把基本功练好，这样才能在不同的需求下运用自如~~。

下面把 Sales Order Number 填充到“Tablix”，如图：

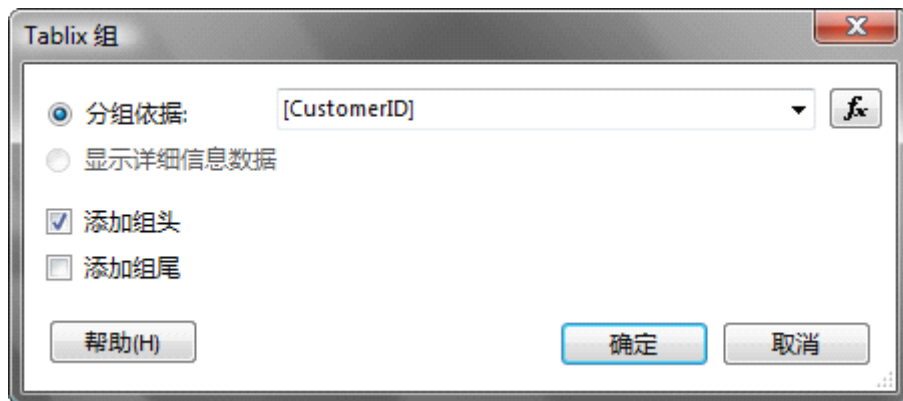
		Sales Order Number	
		[SalesOrderNumber]	

你可以从数据集直接拖入相应的列中，也可以点击文本框边上的小图标来选择字段。或直接打开表达式窗口来编辑。方法很多~~

这里这需要填充该字段，我们把多余的列删除。并添加“行组”，如下图：



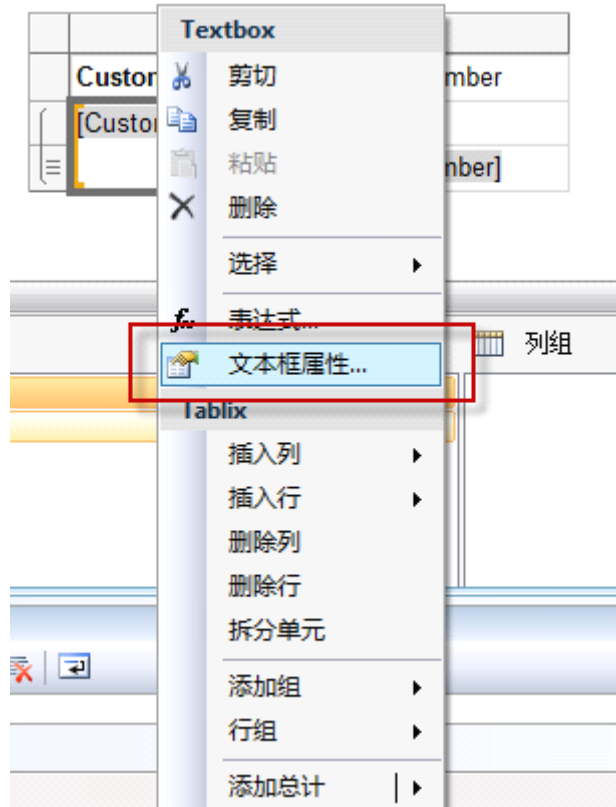
选择“父组”，在弹出的窗口中设置参数如下图：



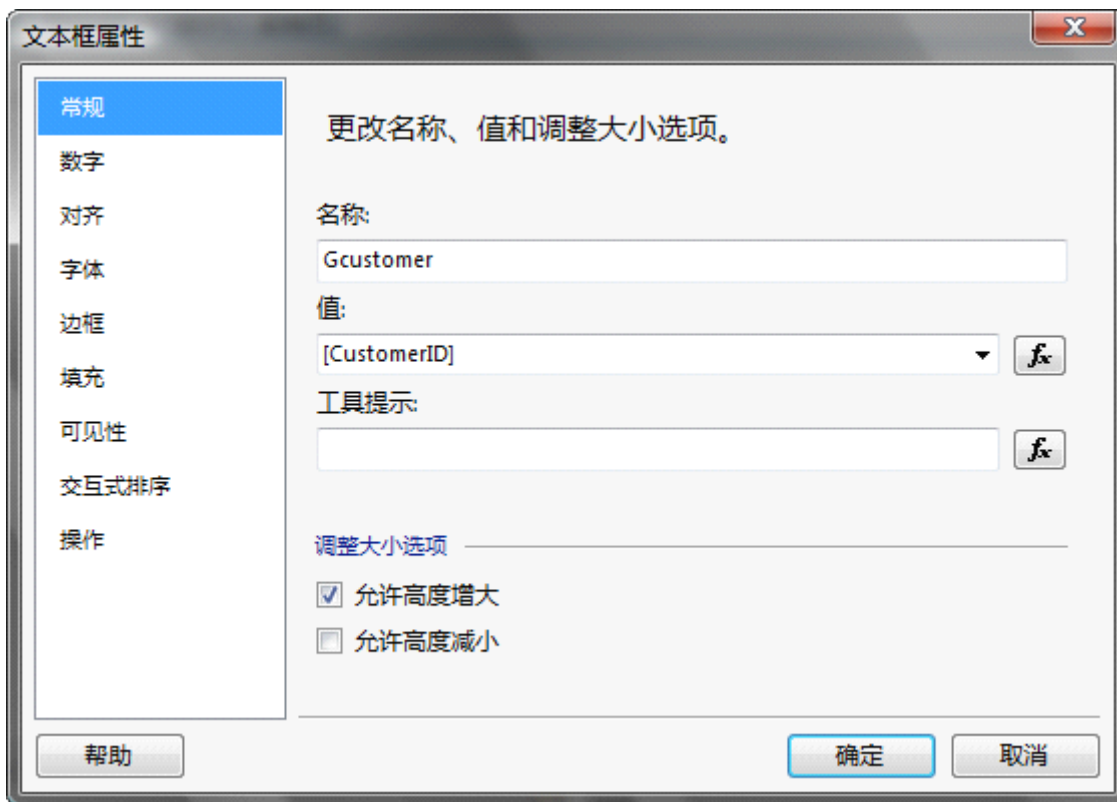
点击“确认”后，应该如下图：列标头可以自行修改。

Customer ID	Sales Order Number
[CustomerID]	
	[SalesOrderNumber]

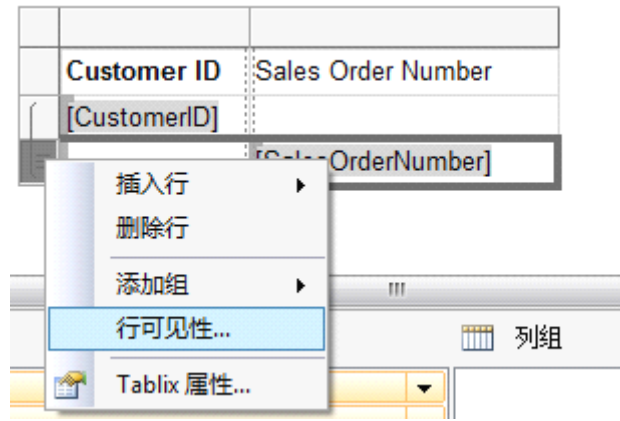
同时修改分组项文本框的名称，我们需要通过该报表项来控制订单的显示。如图：



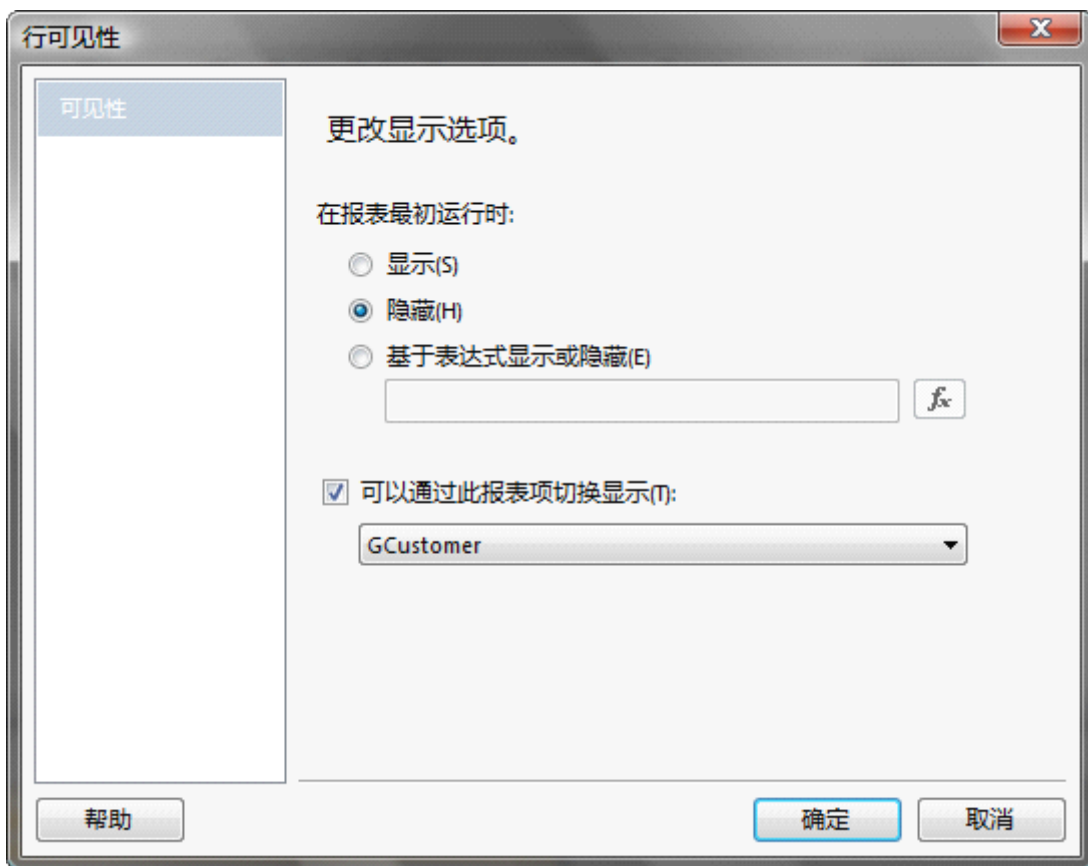
弹出如下窗口：在此设置名称为：Gcustomer。



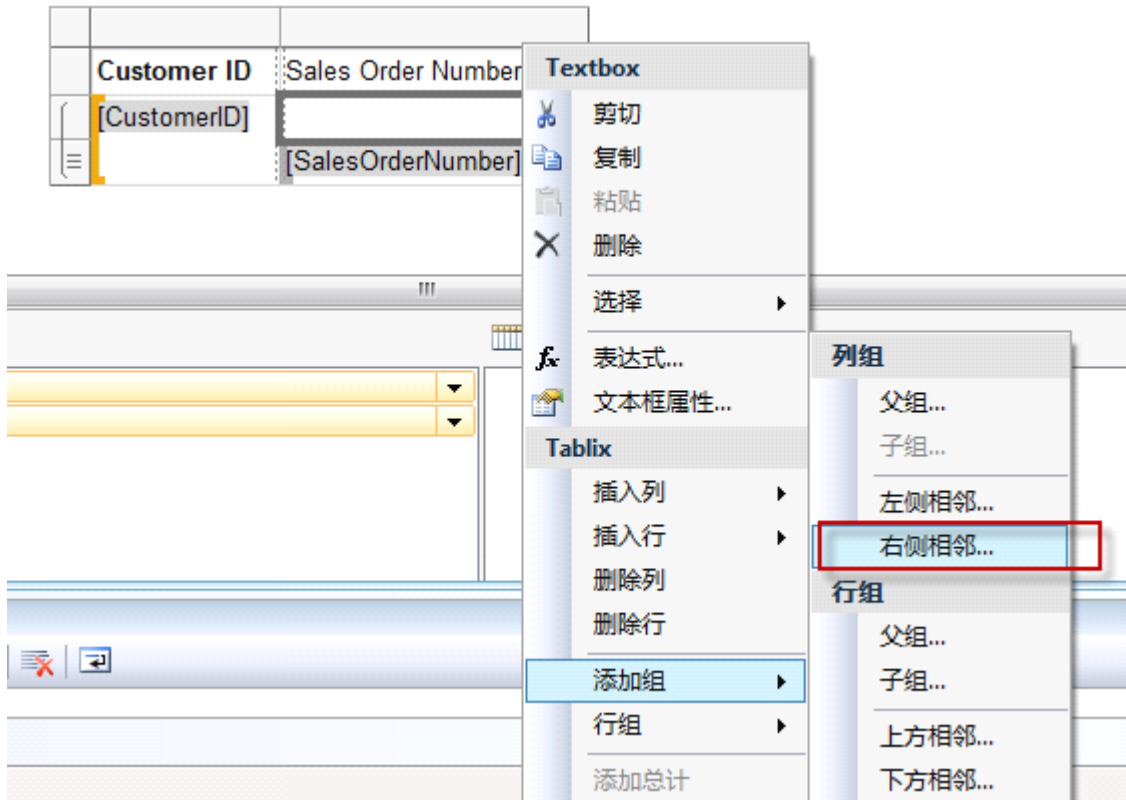
设置好以后，再来设置通过“Gcustomer”报表项来控制数据行的显示，如下图，右键数据行，弹出快捷菜单。选择“行可见性”。



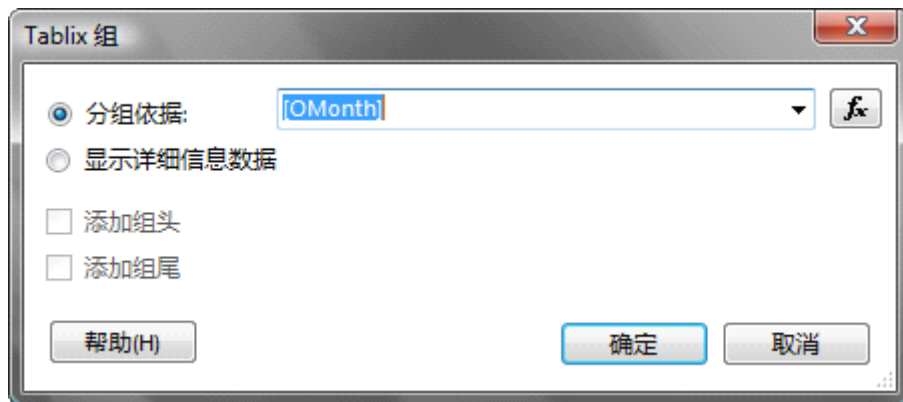
弹出设置窗口：



点击“确定”后，你可以先预览一下，看是否可以控制隐藏，显示。只不过现在看到的是半成品~~。我们继续向下设置：下面应该设置动态列了~，我们希望动态列显示在“Sales Order Number”列的右边。所以右键“Sales Order Number”列的文本框。弹出如下快捷菜单~



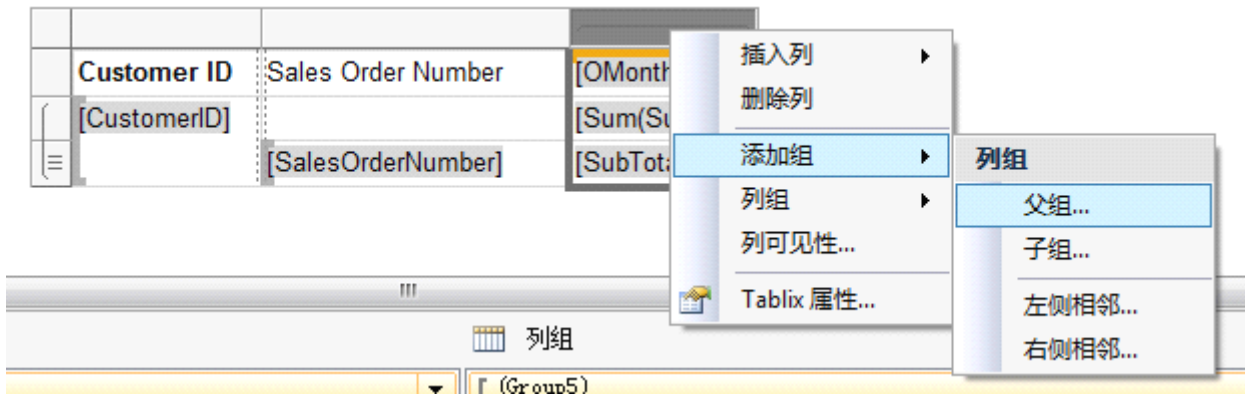
我们选择列组中的“右侧相邻”。



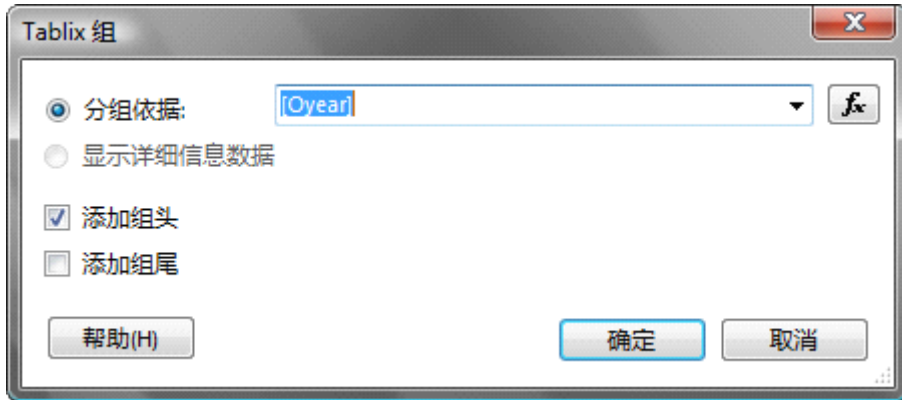
选择月份作为分组依据，点击确定后，多出一列，我们填充上相应的值，在列标题绑定 Omonth 字段，详细信息绑定 SubTotal, Customer ID 分组对 SubTotal, 值进行汇总。填充后如下图：

	Customer ID	Sales Order Number	[OMonth]
[CustomerID]			[Sum(SubTotal, CustomerID)]
		[SalesOrderNumber]	[SubTotal]

下面再对 Omonth 列组添加一个父组年份。右键动态列从菜单中选择“添加组”->“父组”。



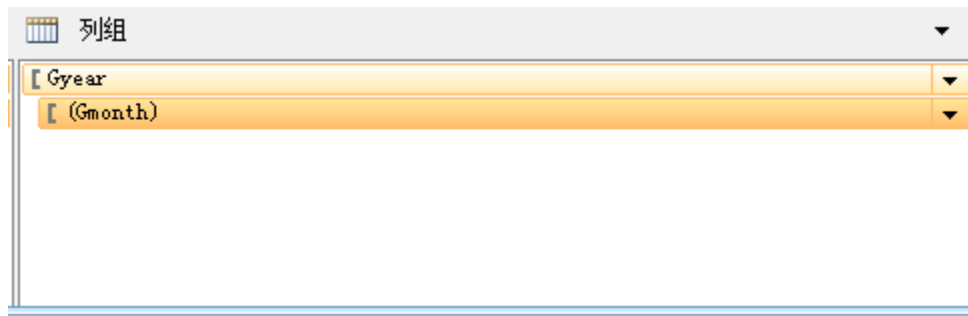
进入设置窗口：设置分组依据为 Oyear



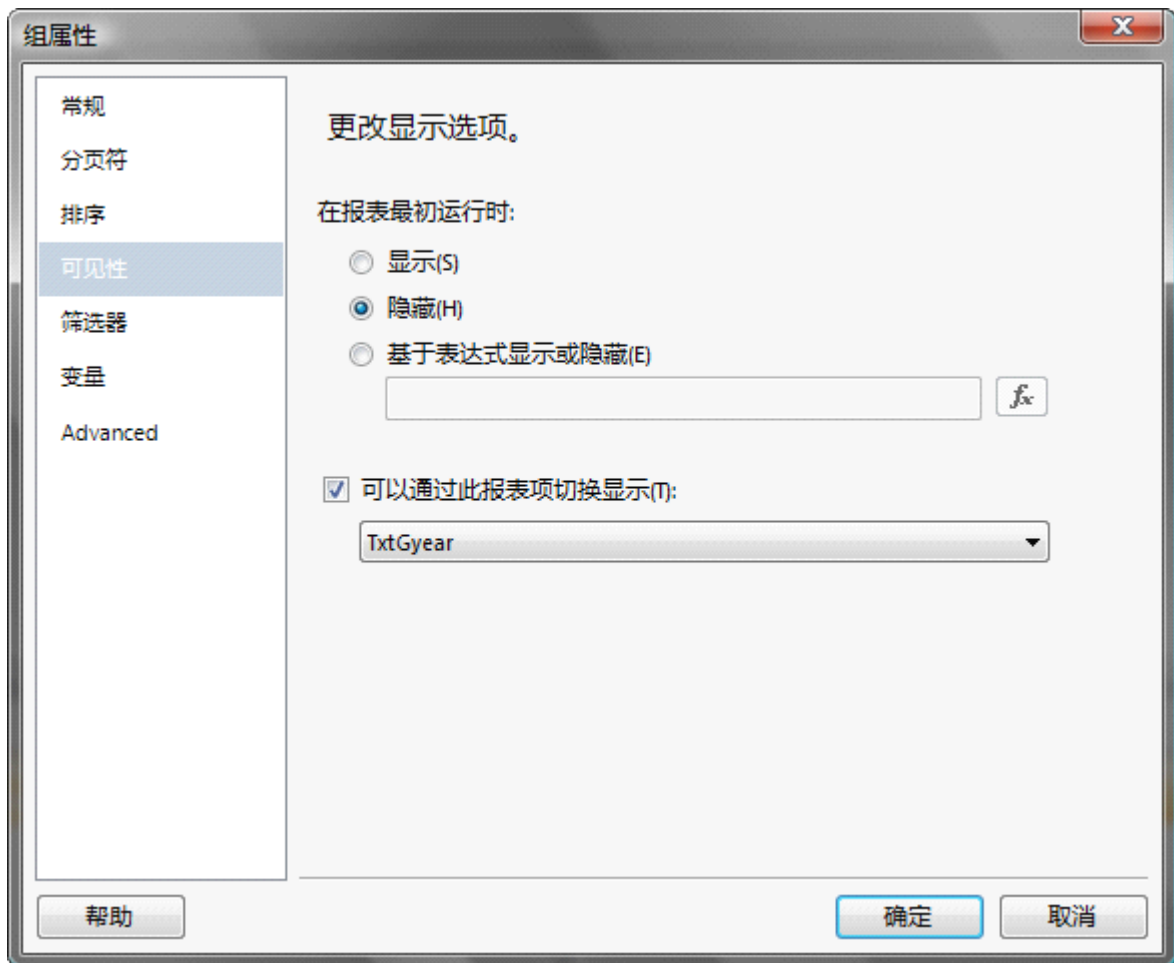
经过如上设置，Tablix 应该如下图：

				[Oyear]
Customer ID	Sales Order Number		[OMonth]	
[CustomerID]			[Sum(SubTotal)]	
	[SalesOrderNumber]		[SubTotal]	

我们再来设置 Omonth 组受控于父组的报表项。我们双击“列组”窗口中的 Gmonth 列组，然后在弹出的窗口中设置可见性。



设置窗口如下：可以先设置好父组的报表项，然后再在这里选择。



OK，设置工作已完成了，我们对字体及对齐方式做调整，让它看起来更美观～～最终如下图：

		[Oyear]	
Customer ID	Sales Order Number	Year Total	[OMonth]
[CustomerID]		[Sum(SubTotal)]	[Sum(SubTotal)]
	[SalesOrderNumber]		[SubTotal]

点击“预览”，看看效果吧～～

Customer ID	Sales Order Number	Year Total	2002			
			7	8	9	10
2		9216.3596		9216.3596		
	SO46976			9216.3596		
3		71112.1316			71112.1316	
4		238031.9559	118394.1258			119637.8301
5		20720.6175			20720.6175	
7		6019.0371	3569.4338			2449.6033
10		45089.9298			45089.9298	
12		79060.5774		79060.5774		
16		68272.1366			68272.1366	
17		10766.8293	8789.9444			1976.8849
18		58420.5745		58420.5745		
20		9047.7586			9047.7586	
21		68309.5811			68309.5811	
22		2889.0273		2889.0273		

和我们前面看到的效果完全一下,功夫不负有心人~~。

本篇内容就讲到这里了,大家一定要熟练运用 Tablix ,该控件在日常工作中经常用到,特别2008中对该控件的功能增强后,完全可以替代矩阵控件。如果想把 Tablix 控件当矩阵来用,只需要删除详细数据,保留汇总数据,得到的结果将和矩阵相同。大家可以自己动手做一下,在此不再阐述~~。

希望本篇介绍的内容,在您的工作中能够提供帮助,下一篇将讲解矩阵,矩形区块,列表,子报表等控件的使用~~