



Report Builder 3.0

实用手册

Report Builder 3.0 实用手册

Report Builder 3.0 是一种可以用于设计和发布自己的报表的报表创作工具，它同微软最新的 SQL Server 2008 R2 可以很好地协作，从而给用户带来更加强大的商业智能功能。在本次技术手册中，我们将对 Report Builder 的使用技巧进行一个系统详尽的总结，其中包含了使用 Report Builder 3.0 创建报表、生成各种图形以及最新最炫的地图报表功能等。

在 Report Builder 3.0 中使用仪表

Report Builder 3.0 包含了许多新的数据可视化特性，它可以让我们生成标准的表和图表之外的报表。这其中的一个特性是仪表(gauge)，它以辐射型或线型格式显示关键性能指示 (KPIs)。

- ❖ 在 Report Builder 3.0 For SQL Server 中创建报表
- ❖ 使用 Report Builder 3.0 for SQL Server 添加仪表
- ❖ 使用 Report Builder 3.0 进行范围配置
- ❖ 设置 SQL Server 报表中的仪表刻度
- ❖ 使用 Report Builder 3.0 配置指针
- ❖ 使用 Report Builder 3.0 添加仪表完结篇

使用 Report Builder 3.0 添加图表

如同仪表一样，图表能通过简明扼要的可视化格式显示有意义的数据来强化 Report Builder 3.0 报表效果。但是，图表能够描述远比仪表复杂的信息，并且图表可以有许多方式来保证数据的正确表示。

-
- ❖ 使用 Report Builder 3.0 添加图表
 - ❖ 在 Report Builder 3.0 上创建条形图——第一部分
 - ❖ 在 Report Builder 3.0 上创建条形图——第二部分
 - ❖ 在 Report Builder 3.0 中生成柱状图和折线图
 - ❖ 在 Report Builder 3.0 中使用区间图
 - ❖ 在 Report Builder 3.0 中使用饼状图
 - ❖ 在 Report Builder 3.0 中使用漏斗图

使用 Report Builder 3.0 进行地理绘图

Report Builder 3.0 的其中一个最令人兴奋的就是新的地图功能，它可以添加由 1 个或多个显示地理和分析数据的层组成的地图。这个地图也可以与 Microsoft Virtual Earth 整合，以实现一个生动的数据可视化工具。

- ❖ SQL Server 2008 R2：使地理绘图更简单
- ❖ 在 Report Builder 3.0 中添加一个多边形图层到地图
- ❖ 在 Report Builder 3.0 中修改多边形图层的属性
- ❖ 完成 Report Builder 3.0 中地图点的创建
- ❖ 用 Report Builder 3.0 添加一个块图层到地图

Report Builder 3.0 中的矩阵

矩阵是一种表，它按行、列或行与列显示汇总数据——类似于交叉表或 数据透视表。在列组或行组中的唯一行的数量决定了矩阵报表返回的行和列的数量。

- ❖ Report Builder 3.0 矩阵初探
- ❖ Report Builder 3.0 创建矩阵报表：准备工作
- ❖ Report Builder 3.0 创建矩阵报表：添加组层次

-
- ❖ Report Builder 3.0 创建矩阵报表：添加总数到矩阵
 - ❖ Report Builder 3.0 中矩阵配置技术

在 Report Builder 3.0 For SQL Server 中创建报表

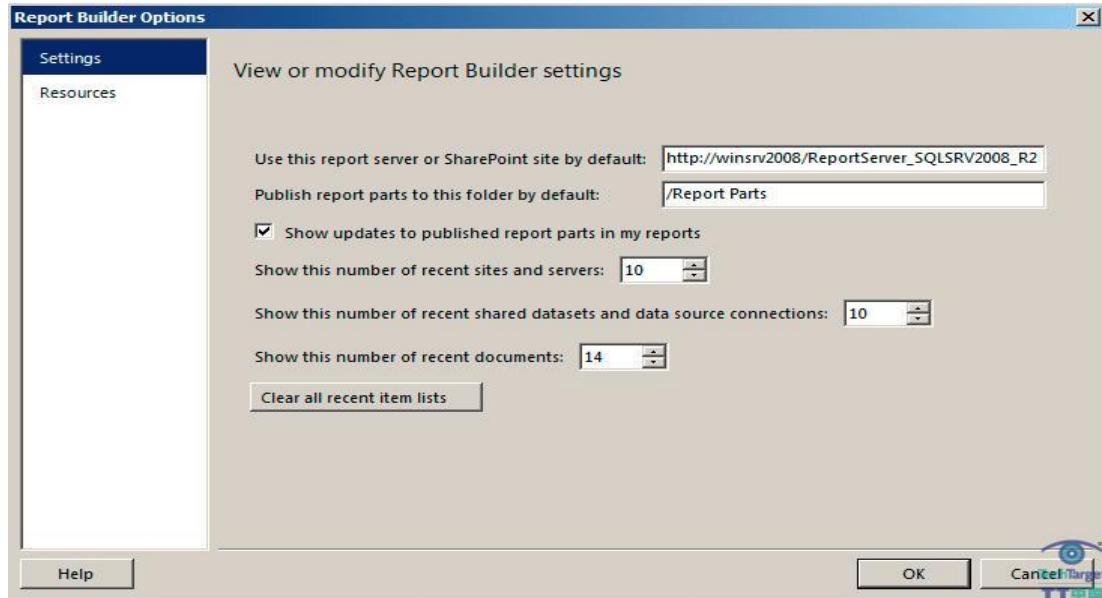
Report Builder 3.0 是一个独立报表制作工具，可以用来设计和定制报表，并将它们发送到一个报表服务器上。虽然这个工具是 SQL Server 2008 R2 的一部分，我们也可以单独下载 Report Builder，并将它指向 SQL Server Reporting Services (SSRS) 报表服务器，其中报表在上面就像其它的 SSRS 报表一样以相同方式生成。

Report Builder 3.0 包含了许多新的数据可视化特性，它可以让我们生成标准的表和图表之外的报表。这其中的一个特性是仪表(gauge)，它以辐射型或线型格式显示[关键性能指示 \(KPIs\)](#)。

本文我将说明如何添加一个仪表到一个报表，并配置该仪表的属性以显示一个有重要信息的 KPI。

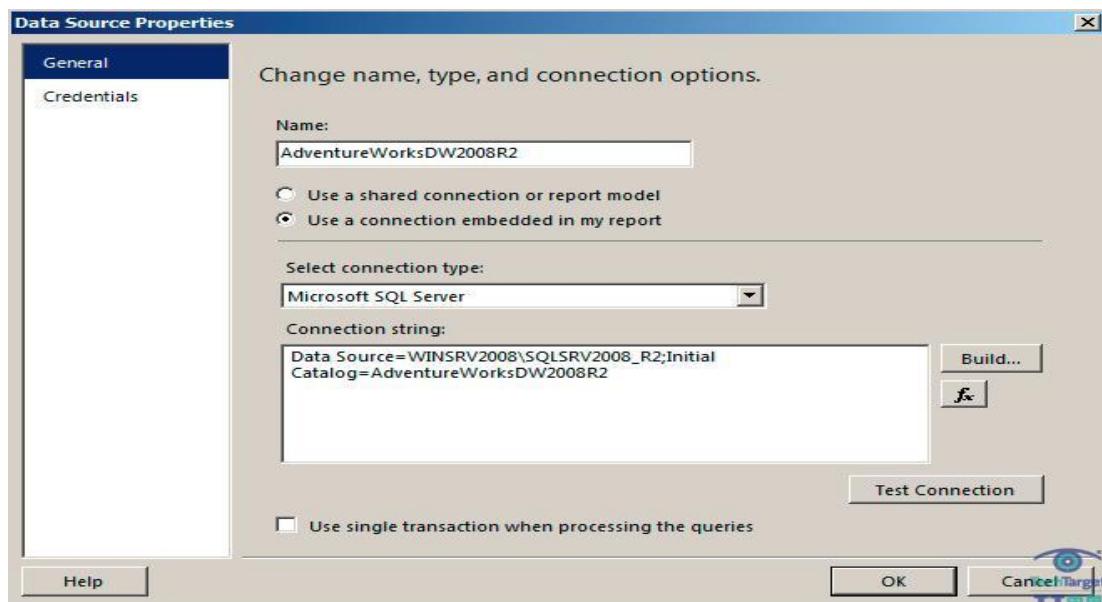
创建报表

为了说明如何添加一个仪表到一个 Report Builder 报表，我创建[SQL Server 2008 R2 的 2009 年 11 月社区技术预览 \(CTP\)](#)的一个本地实例。该安装包括数据引擎和 Reporting Services。我同时也在相同的服务器上安装了 Report Builder 3.0 的 11 月 CTP 版本。接下来，我配置 Report Builder 的选项，这样它指向 SSRS 报表服务器，如图 1 所示。

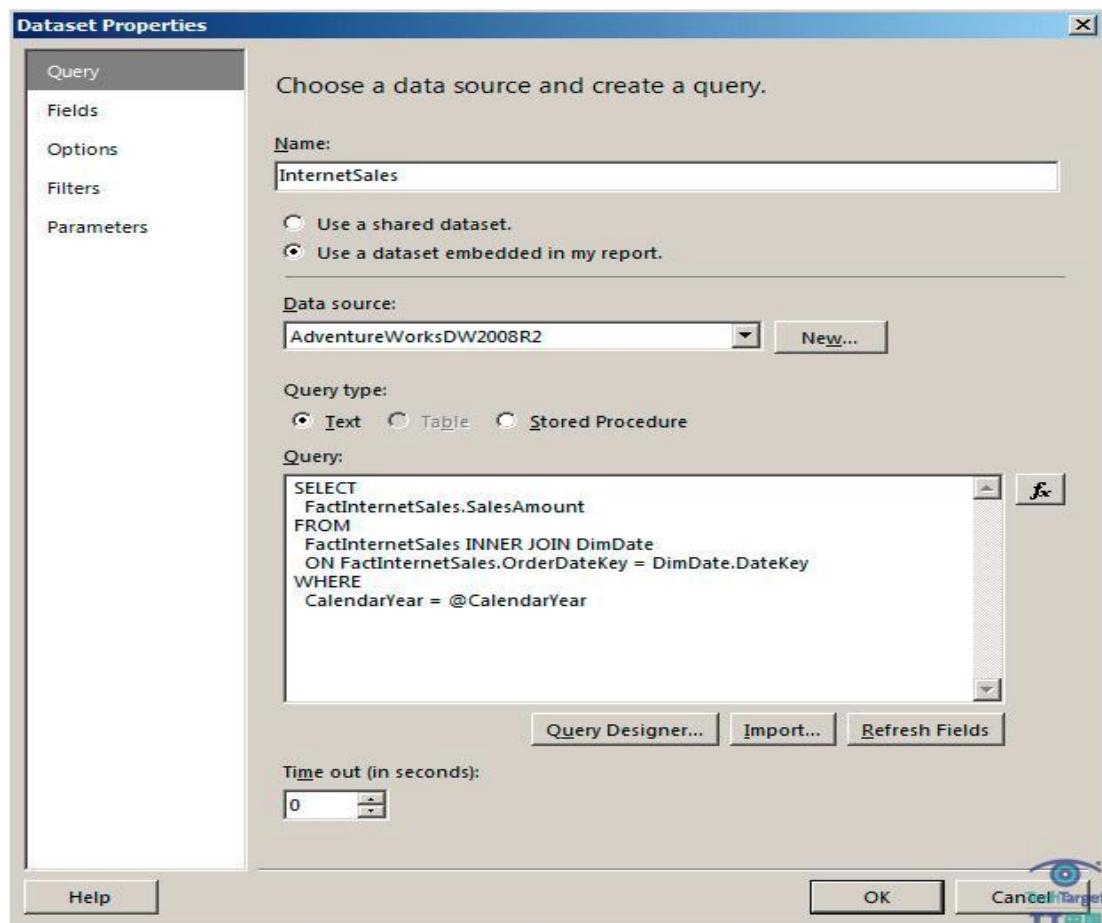


注意，一旦 Report Builder 指向了一个报表服务器，我们就可以开始创建报表。我们所创建(和保存)的任何报表都会自动地提交到报表服务器上。

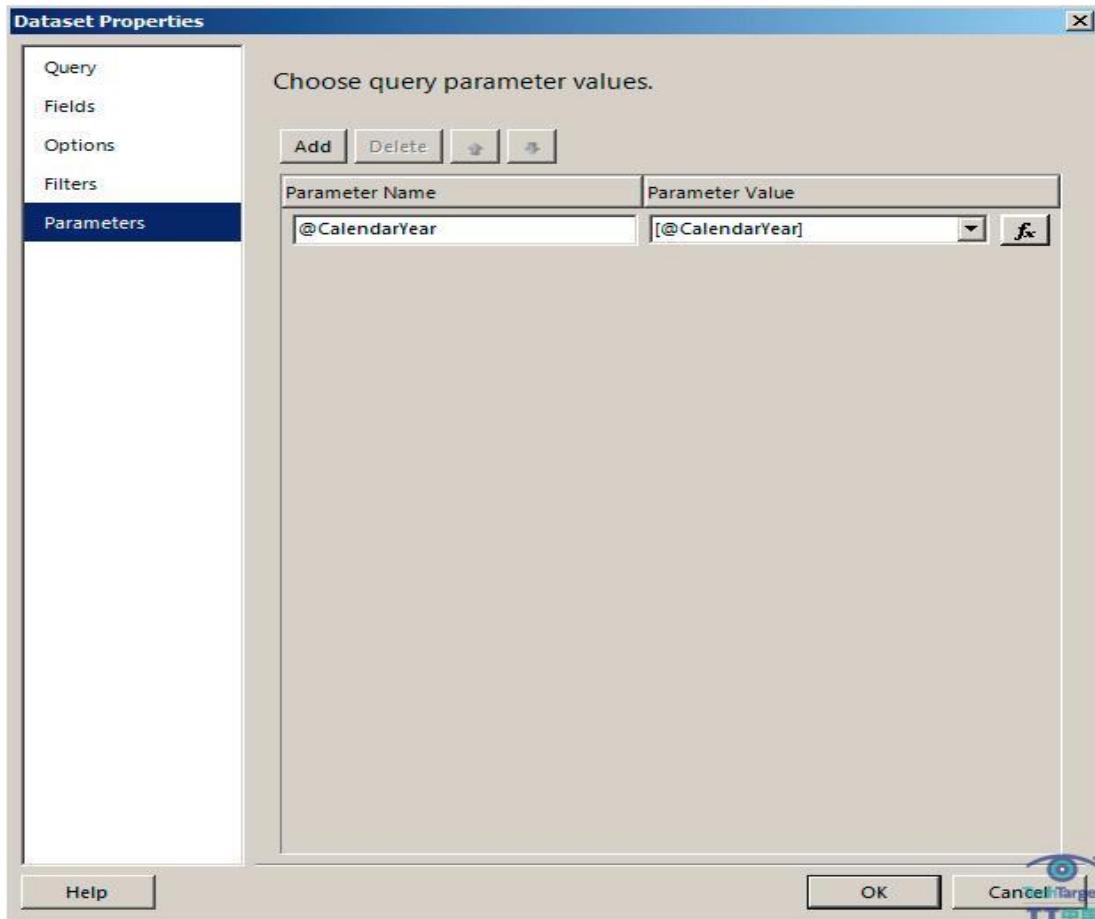
在我创建了 SQL Server 和 Report Builder 后，我再安装适用于 SQL Server R2 的 AdventureWorks 示例数据库(我们可以从 CodePlex 下载到该[数据库安装文件](#))。接下来，我在 Report Builder 上配置了一个数据源，它指向 AdventureWorksDW2008R2 数据库。图 2 显示了 AdventureWorksDW2008R2 数据源的属性。



当数据源配置好后，我创建 InternetSales 数据集，它查询每一个 Internet 销售的数目。图 2 显示了 InternetSales 数据集的 Dataset Properties 对话框，它包括了用于从数据源查询数据的 Transact-SQL 语句。



注意图 3 显示的 SELECT 语句包含了一个 WHERE 子句，它将变量@CalendarYear 的值赋给字段 CalendarYear 。由于有了这个子句，用户在查看报表时将能够指定一个年份。这个变量将动态地添加到 Dataset Properties 对话框的 Parameters 页，如图 4 所示。



注意在第一次查看 Parameters 页时，其中的参数值可能为空，但如果关闭该对话框然后再打开，就会有值出现。

当数据源和数据集创建好后，我们就可以添加一个仪表到一个报表上。注意，通常一个报表会包含许多其它特性。然而，基于本文的写作目的，我这里只介绍如何添加仪表。仪表将会提供一个 KPI，它反映 Internet 销售总数，它们是存储在 AdventureWorks 数据仓库中的。

注意：关于如何配置 Report Builder 选项和如何创建数据源和数据集的详细步骤，可以参考 Report Builder 2.0 Help，它可以从 Report Builder 中打开。

(作者: Robert Sheldon 译者: 曾少宁 来源: TechTarget 中国)

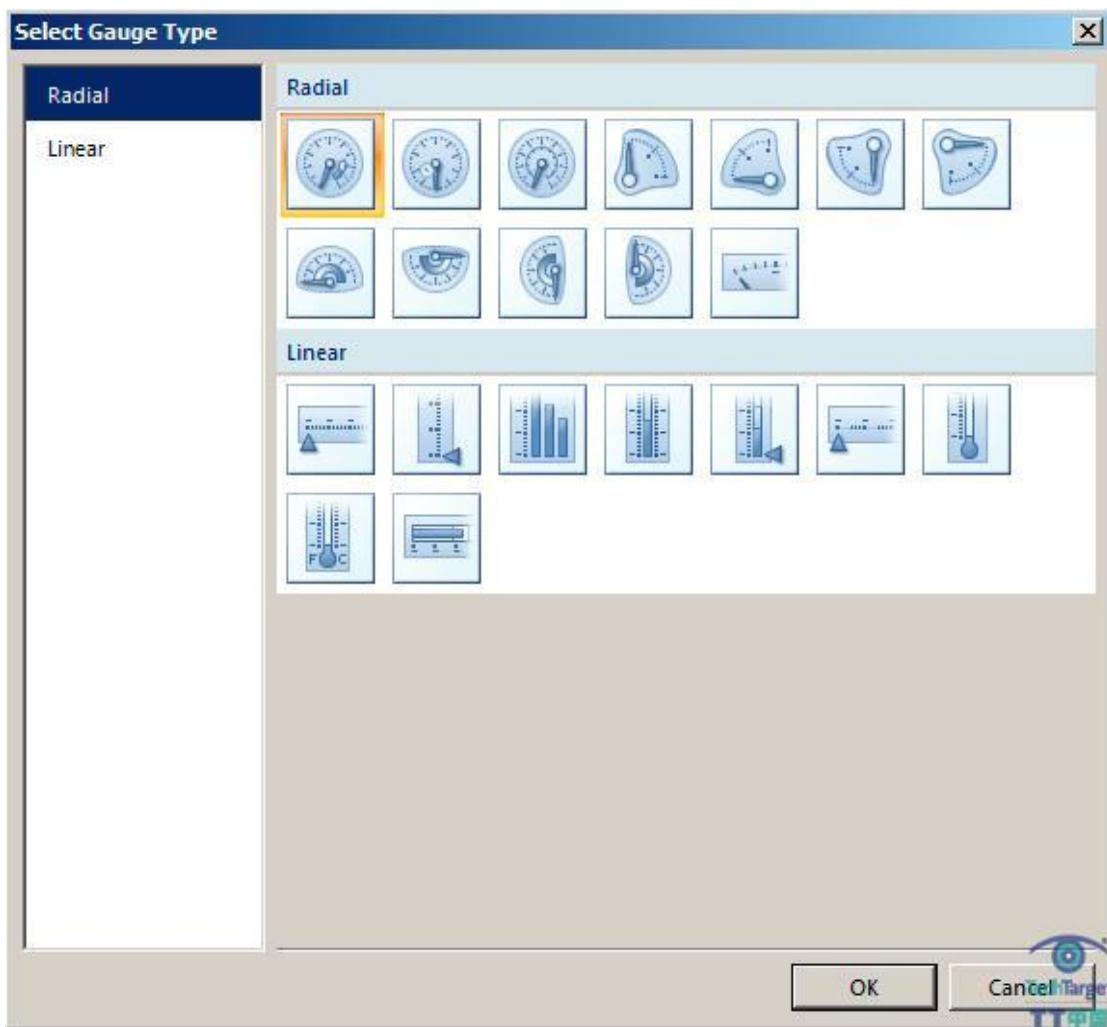
原文标题：在 Report Builder 3.0 For SQL Server 中创建报表

链接：http://www.searchdatabase.com.cn/showcontent_30985.htm

使用 Report Builder 3.0 for SQL Server 添加仪表

添加仪表到报表

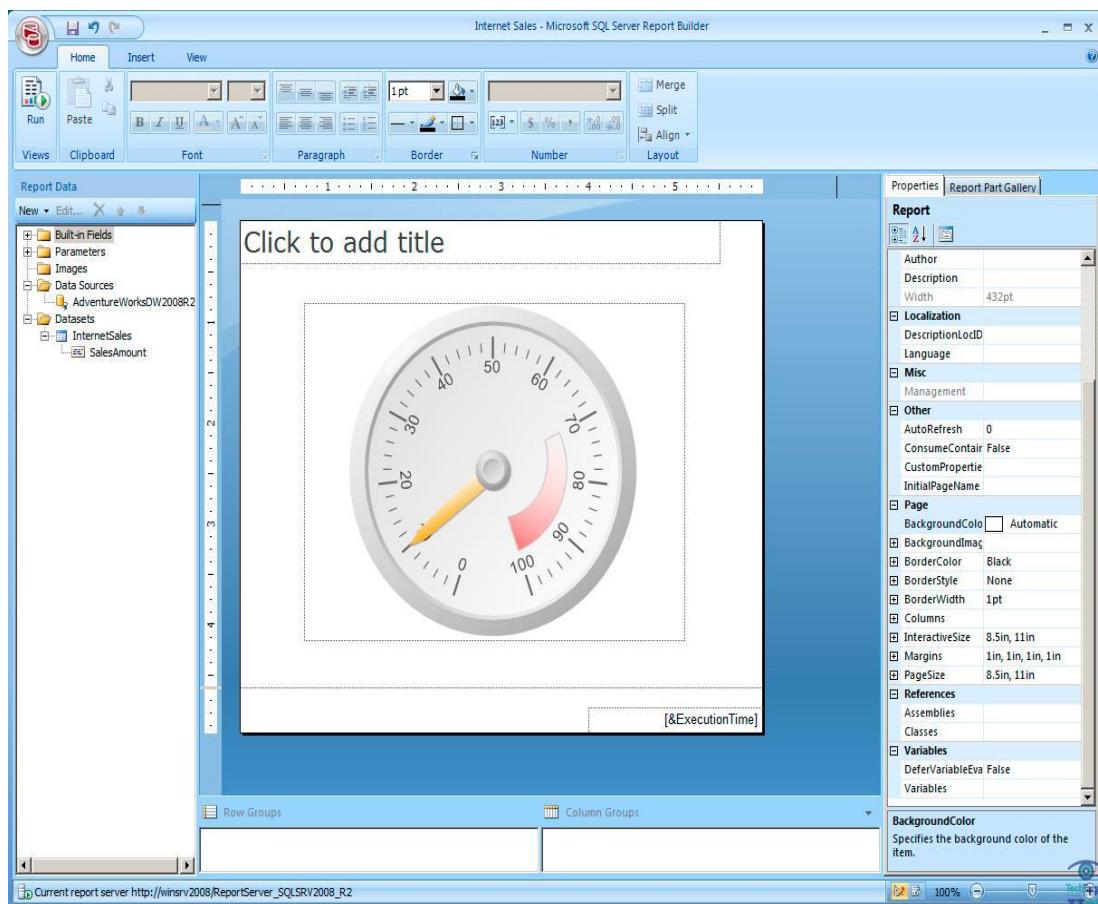
为了添加一个仪表到一个报表，选择菜单条的“Insert”，然后双击“Gauge”图标。然后就会显示 Select Gauge Type 对话框，如图 5 所示。



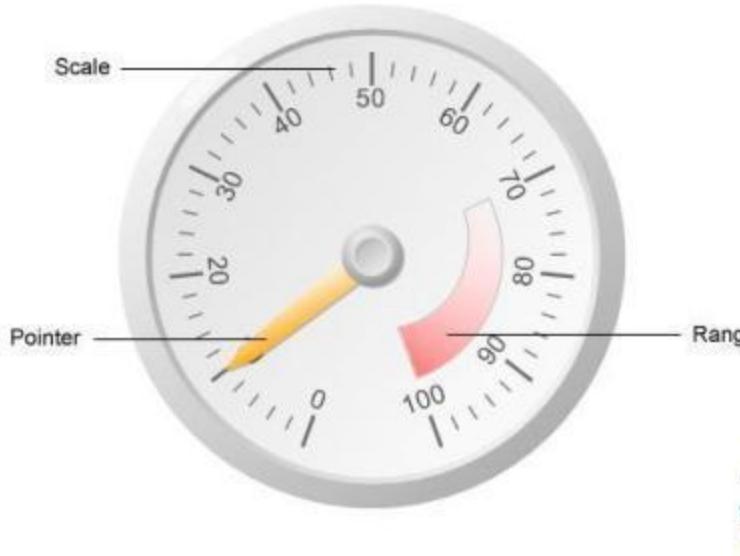
“Select Gauge Type”对话框是选择插入到报表的仪表类型的地方。Report Builder 3.0 支持 2 种类型的仪表：辐射型和线型。这两种类型的不同点基本上就是形状，

我们可以在对话框中看到仪表的缩略图的。添加一个仪表到报表上，只需要双击其中一个缩略图就行的。然后这个仪表会被添加到报表的设计界面中，我们可以对它进行缩放和设置位置。

对于这个练习，我选择 Select Gauge Type 对话框的第一个仪表。在我添加该仪表到设计界面后，我设置了它的大小和位置，如图 6 所示。



在添加仪表到报表后，我们可以配置它的属性。这些属性分成不同的类型，分别与组成该仪表的组件相对应。其中最常配置的 3 个组件是指针、刻度和范围，如图 7 所示。



在第 2 部分，我们将更详细地学习这些组件，这样我们可能更好地理解它们的工作方式和如何配置它们的属性。

(作者: Robert Sheldon 译者: 曾少宁 来源: TechTarget 中国)

原文标题: 使用 Report Builder 3.0 for SQL Server 添加仪表

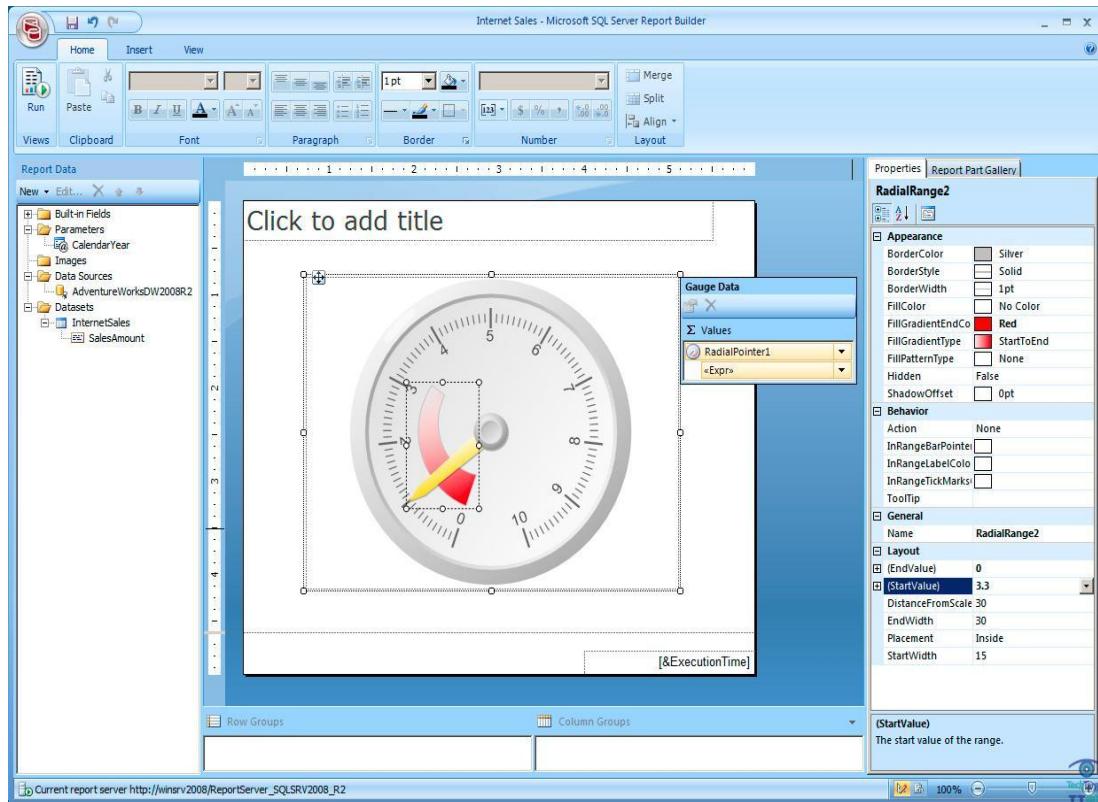
链接: http://www.searchdatabase.com.cn/showcontent_30987.htm

使用 Report Builder 3.0 进行范围配置

一个范围标记了刻度的一个特定区域，它有助于突出特定的值。例如，在一个间隔为 1 到 10 的刻度中，我们可能会考虑添加一个从 8 到 10 的范围来突出这些目标数值。

然而。要记住这些范围是专门绑定到刻度的间隔值的。这意味着如果我们修改了间隔值，将会影响到所定义的所有范围。例如，当我修改了刻度间隔 MaximumValue 的值为 10，那么定义在仪表上的默认范围就会消失。我们可以在配置刻度之前配置范围来解决这个问题，首先删除范围或者不管它——但是要知道原始的范围已经定义在仪表中了。它只是没有显示出来而已。

在本练习中，我保留原始的范围，并添加了 2 个新的范围。第 1 个范围标识刻度的最低值，第 2 个标识最大值。添加一个范围，可以右击仪表，然后选择“New Range”。当新的范围添加到仪表中后，我们就可以在“Properties”面板或“Radial Scale Range Properties”对话框中配置范围的属性。图 12 显示的是我添加的范围，它显示在“Properties”面板中。

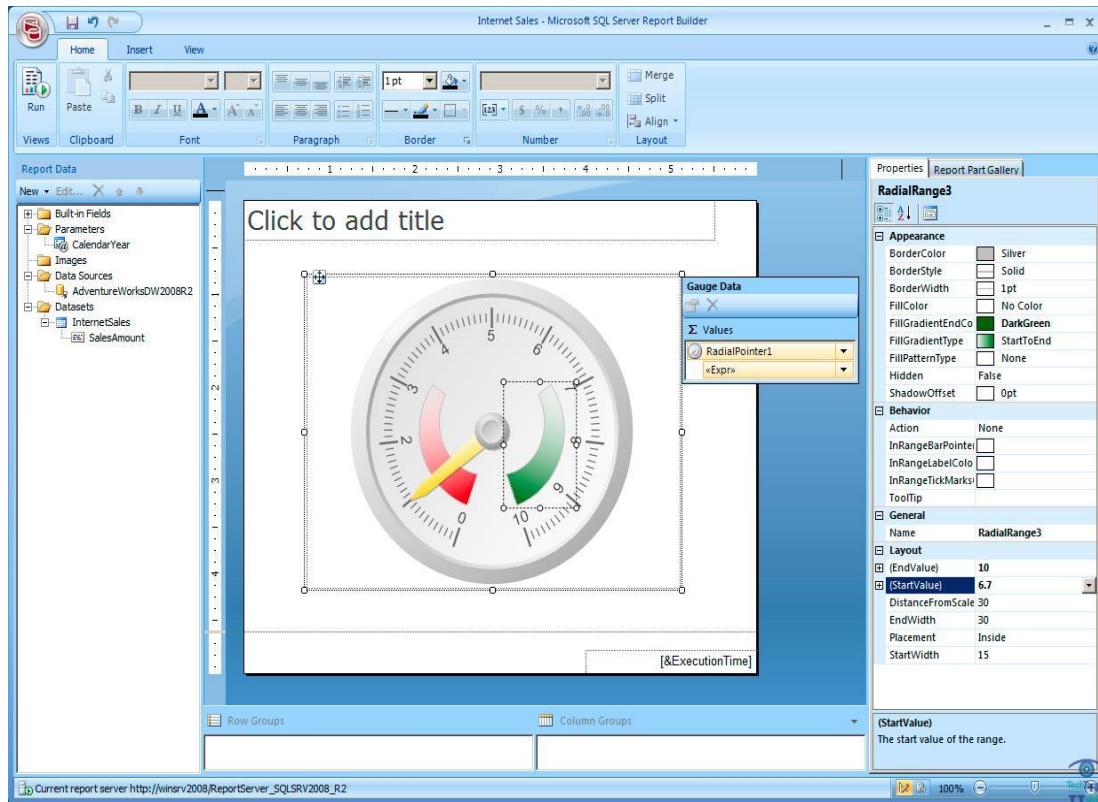


对于新的范围，我设置了以下 3 个属性的值：

- FillGradientEndColor: Red
- EndValue: 0
- StartValue: 3.3

我选择 3.3 作为开始值，因为我知道数据仓库将会返回的值是多少。本练习中我选择的数值能帮我演示一些 Report Builder 3.0 的特性。当然，我们是可以自由地根据需要来设置范围值。

在我添加了第 1 个范围后，我又添加了第 2 个范围，如图 13 所示。



对于第 2 个范围，我设置了以下属性的值：

- FillGradientEndColor: DarkGreen
- EndValue: 10
- StartValue: 6.7

对于两个范围，这些都是我所配置的仅有的 3 个属性。对于其它属性，我保留它们的默认值。我们也可以根据具体报表的需要来配置范围的属性。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 使用 Report Builder 3.0 进行范围配置

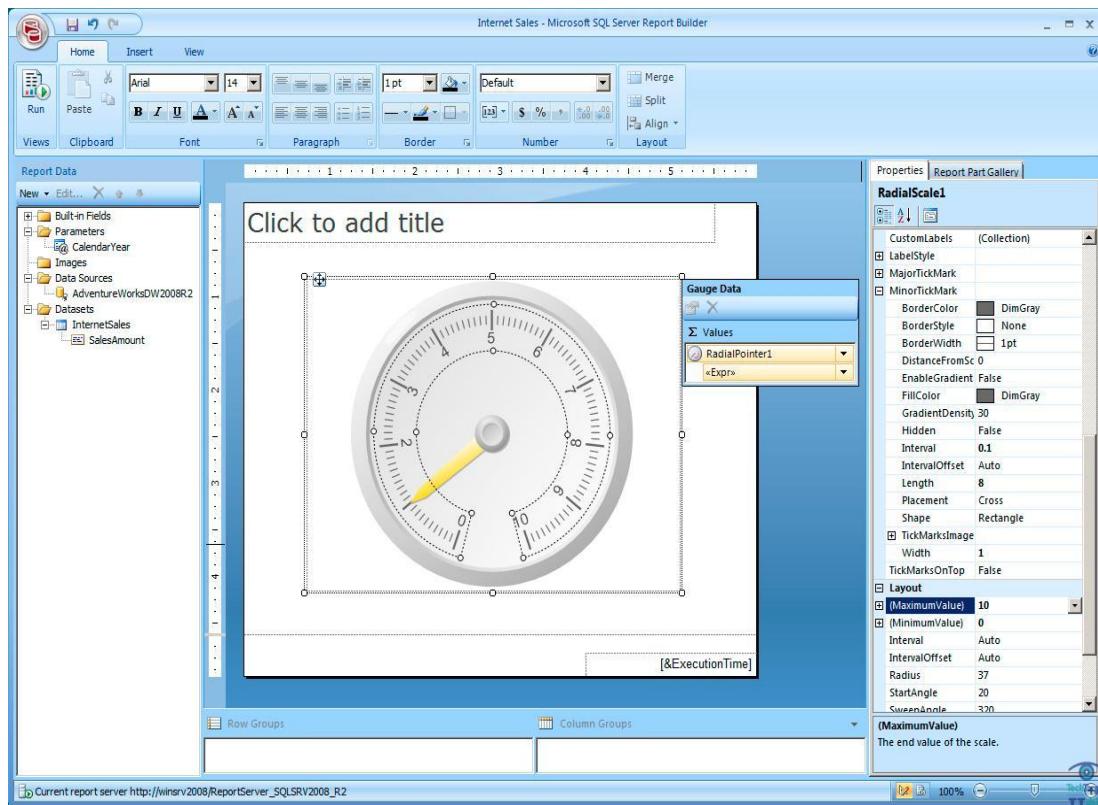
链接: http://www.searchdatabase.com.cn/showcontent_31053.htm

设置 SQL Server 报表中的仪表刻度

配置好指针后，我们就可以配置刻度了。（注意不是一定要按照我这里的顺序来配置每一个组件。你可以根据你的需要来配置它们。）

刻度表示用于比较指针值的标准。例如，一个仪表的默认刻度是从 0 到 100 的。如果 Internet 销售有 2020 万美元，那么指针将指向 20.2(百万)的刻度位置。

它表示的是，在 AdventureWorks 数据仓库中每一年的 Internet 销售大约在 320 万美元到 980 万美元之间。所以根据本练习的要求，让我们把刻度设置为 0 到 10(百万)。设置这个刻度，可以点击选择该刻度，然后在“Properties”面板中修改属性。我们也可以右击该刻度，然后点击“Scale Properties”来打开“Radial Scale Properties”对话框。图 11 显示了在“Properties”面板中的属性。



注意“MaximumValue”属性在“Properties”面板中是高亮显示的。默认情况下，这个值是被设为 100 的，但是我将它修改成 10，这样刻度的最大值就变成 10 了。然后我们可以保留“MinimumValue”属性值为默认的 0。注意现在仪表中的刻度范围是 0 到 10。

如果返回图 11 中的“Properties”面板，我们将看到 MinorTickMark 分组已经展开，这样分组中的所有属性都显示出来。(这个分组是对应于“Radial Scale Properties”对话框的 Minor Tick Marks 页面。)一个刻度上小刻度指的是数字之间的标记，这些数字是主刻度标记。例如，图 11 中仪表的刻度有从 0 到 10 的主刻度数，以及这些数字之间的小刻度标记。

类似于仪表的其它组件，Report Build 允许我们控制刻度标记的显示。在本例中，我设置了小刻度标准的“Interval”属性为 0.1，这样销售中每 10 万(1 百万的十分之一)会对应一个小刻度。

“MaximumValue”和“Interval”属性是我对刻度所修改的两个属性。其余属性我保留它们的默认值。我们也可以根据需要修改任何一个属性，以保证报表有一致的外观。我们也可以添加额外的刻度。再一次，我建议大家练习使用这些属性，这样有助于理解它们的使用方法和他们的可能值。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

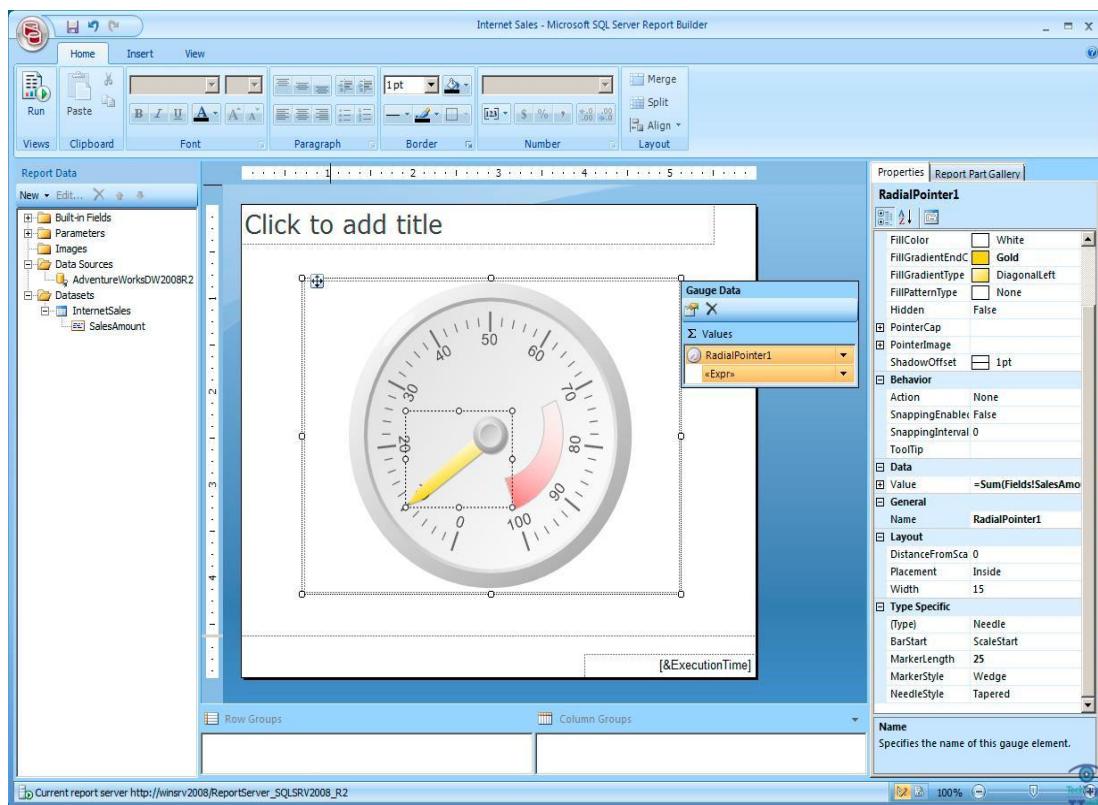
原标题: 设置 SQL Server 报表中的仪表刻度

链接: http://www.searchdatabase.com.cn/showcontent_31045.htm

使用 Report Builder 3.0 配置指针

指针——顾名思义——指的是与配置到该指针的值一致的刻度数值。例如，我添加到这个报表的仪表是为了跟踪 Internet 销售的。所以，指针将指向匹配 Internet 销售总数的刻度数值。

配置指针，在设计界面上点击指针并保证选中它（我们必须点击它两次。）当该指针，或者其它任何组件被选中时，它会被点虚线所包围。图 8 显示的是一个被选中的指针。注意，当选中指针后，“Gauge Data” 框就会出现，指针的属性也将显示在设计界面右边“Properties” 面板上。



在配置指针属性时，我们必须从数据集中选择决定指针值的域。在这里，这个域是 SalesAmount。如果想要对源字段的值进行汇总（求和或计算行数），我们只需要从 Gauge

Data 框底部的下拉列表中选择该域。然后 Report Builder 3.0 会自动地创建汇总数据的表达式。例如，如果要选择 SalesAmount 域，Report Builder 将会创建以下的表达式：

```
=Sum(Fields!SalesAmount.Value)
```

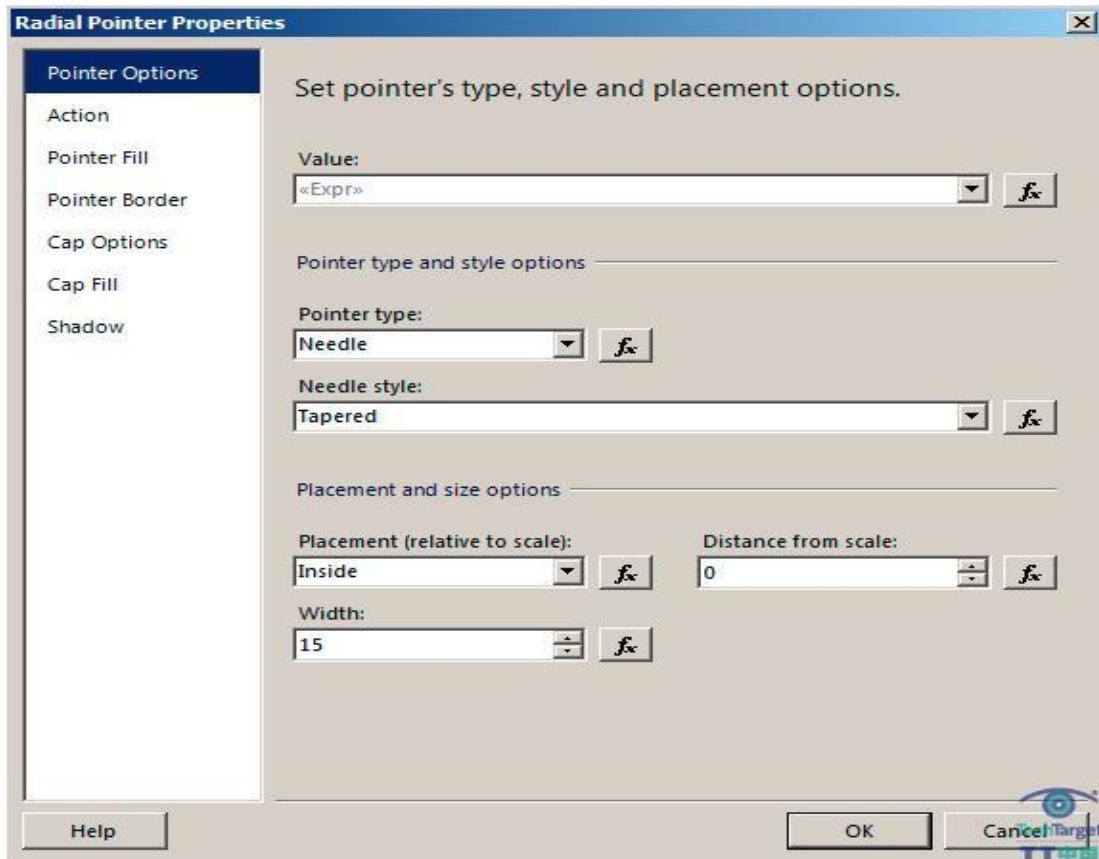
这个表达式可以在“Properties”面板的“Value”中查看。因为 SalesAmount 域包含了数字值，“Sum”方法是用来求和的。然而，如果其中的域不是一个数字值——如，字符串，Report Builder 将会使用“Count”方法来计算行数。

为 SalesAmount 生成的表达式将会返回 Internet 销售总数，这是百万美元的数值。然而，为了使这个数字更具可读性，它将会被简化成一个基数。例如，显示成 6.5 百万美元会比 6,530,343.52 美元简单得多。对于仪表中的汇总数据，我们要保证这些结果数字能够很简便地被读懂。

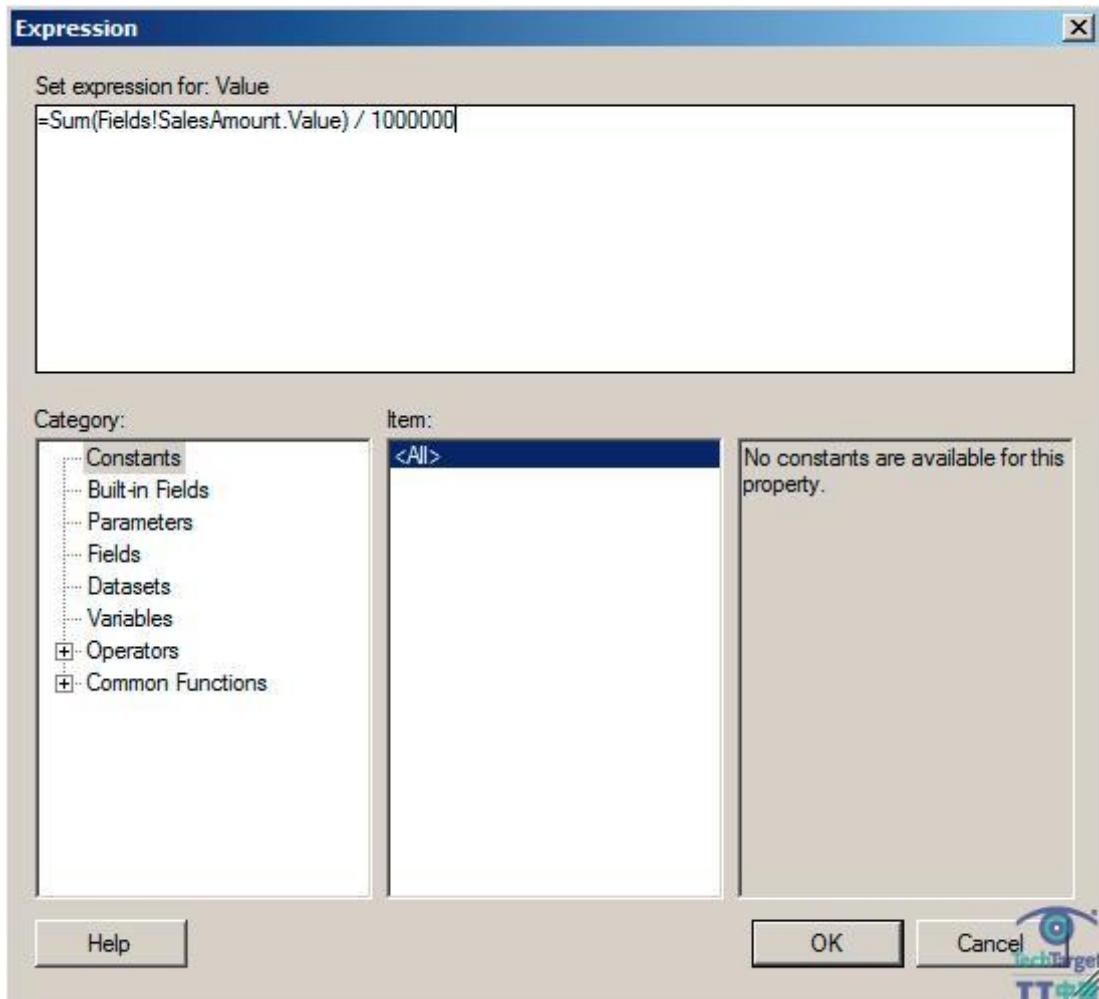
在我们的 Internet 销售报表中，最简单的方法是将销售总数除以 1 百万，然后再相应地调整其刻度。所以，我们先来修改 Value 表达式。因为这个修改很简单，所以我们可以直接在“Properties”面板中进行，将表达式修改成：

```
=Sum(Fields!SalesAmount.Value) / 1000000
```

除了使用“Properties”，还有另一个查看和修改指针属性的方法就是右击指针，然后点击“Pointer Properties”打开“Radial Pointer Properties”对话框，如图 9 所示。



注意 Value 属性包含`<>`值，这是一个属性定义了表达式(而不是默认值)时使用的值。因为我已经修改了该值，所以这里显示的是`<>`。想要查看该表达式，可以点击文本框右边的表达式按钮。这就打开“Expression”对话框，如图 10 所示。



注意这里显示了所修改的表达式。“Expression”对话框对于创建复杂的表达式很有用。从这里，我们可以查看可用的域、数据集、变量、方法和其它语言元素。关于创建表达式的更多信息，请查看 Report Builder Help。

我所修改的另一个指针属性是 FillGradientEndColor，这将它设为金色(Gold)，而不是默认的橙色(Orange)。我们可以使用“Radial Pointer Properties”对话框或“Properties”面板来设置任何一个指针属性。甚至我们还可以添加另一个指针。我建议大家都练习使用每一个属性，以保证你添加到你的报表中的所有仪表具有一致的外观。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

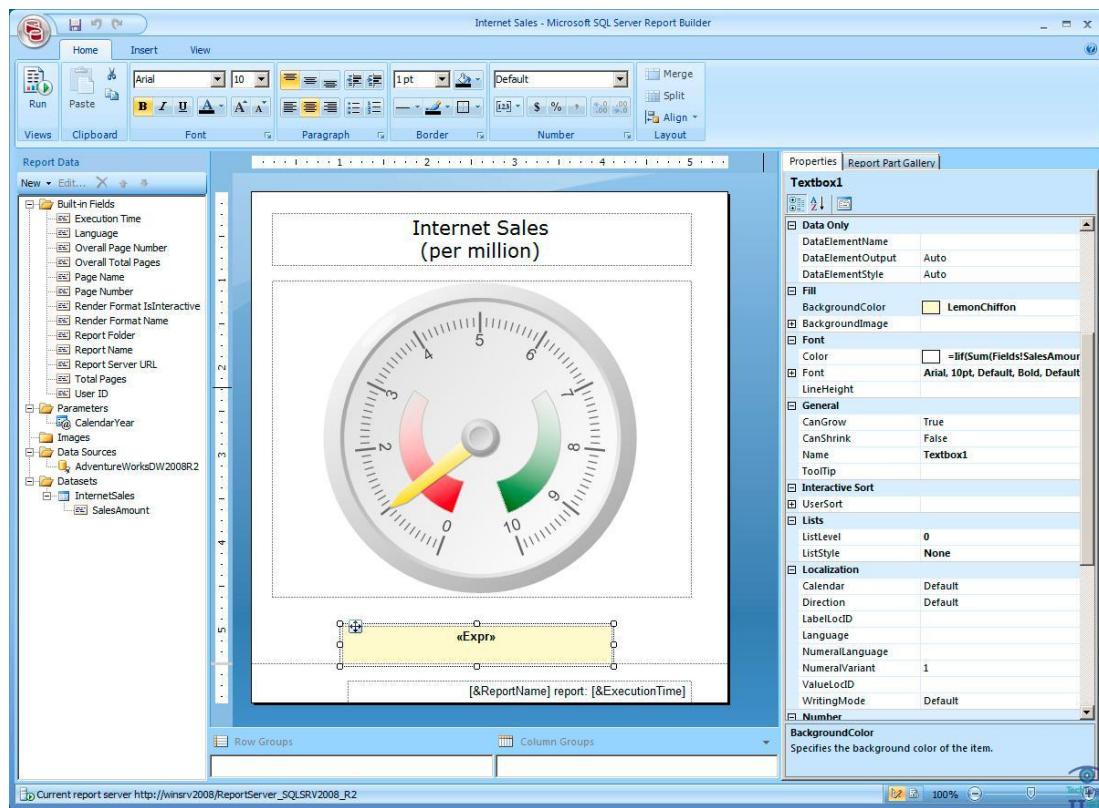
原文标题: 使用 Report Builder 3.0 配置指针

链接: http://www.searchdatabase.com.cn/showcontent_31043.htm

使用 Report Builder 3.0 添加仪表完结篇

首先，我们学习了如何在 [Report Builder 3.0 中创建一个仪表](#)。然后第二部分阐述 [指针、刻度和范围在优化报表的作用](#)。我们几乎完成了添加仪表的学习，但是我们还要学习一些可以用来使报表更有效的技巧。

当我们在 Report Builder 3.0 中配置好了指针、刻度和范围后，我们的登记表就可以使用了。然而，我又添加了更多的元素到下面的报表中，以使它有更佳的外观。图 14 显示的就是附加的组件。



你可能注意到的第一个东西就是给报表设置了一个标题、使文本框居中并调整了字体大小。理所当然地，我们可以给报表设置一个我们喜欢的标题并以特定的格式显示它。Report Builder 3.0 在这方面是非常灵活的，它可以非常自由地实现可视化编辑。

我对报表所作的另一个修改是更新了页脚。我在内置的域 ReportName 旁边加上了一些文本。页脚文本框的表达式现在是这样的：

```
[&ReportName] report: [&ExecutionTime]
```

我们可以在设计界面的左侧的“Report Data”面板上找到一系列内置的域(如图 14 所示)。现在当报表渲染后，报表的名称和时间截会显示。最后，我添加了一个文本框到报表上，对它设置了边框，并将它的“Color”属性设为 LemonChiffon，这显示在图 14 的“Properties”面板上。然后我添加了下面的表达式到字体的“Color”属性中：

```
=Iif(Sum(Fields!SalesAmount.Value, "InternetSales") < 3300000, "Red",  
Iif(Sum(Fields!SalesAmount.Value, "InternetSales") < 6700000, "Black",  
"DarkGreen"))
```

这个表达式规定如果总销售量小于 330 万美元，字体颜色应该是红色的。如果销售量小于 670 万美元(但不小于 330 万美元)，字体颜色应该是黑色的。否则，字体颜色就是深绿色的——即，如果销售量不小于 670 万美元。

除了使用一个表达式来确定字体颜色，我还在文本位置使用了下面的表达式：

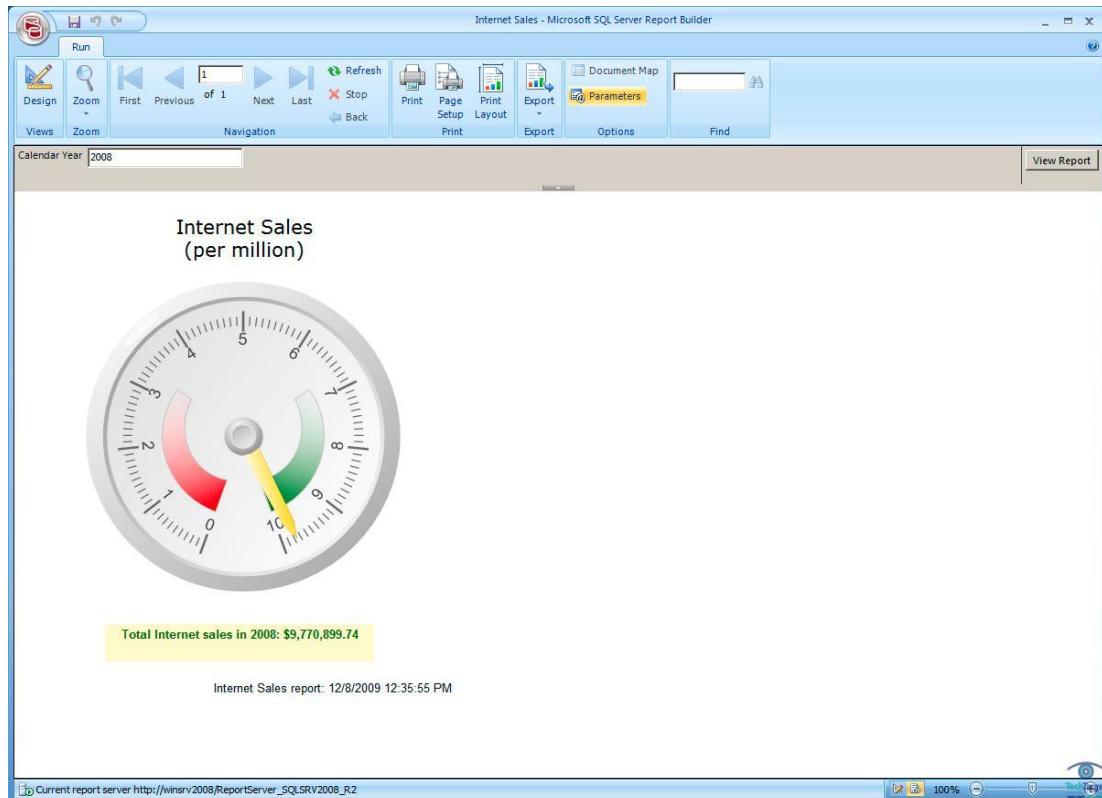
```
= "Total Internet sales in " & Parameters!CalendarYear.Value & ":" &  
FormatCurrency(Sum(Fields!SalesAmount.Value, "InternetSales"))
```

该表达式将参数“CalendarYear”的值和销售总量插入到文本中。例如，如果年份是 2005，那么报表会显示类似于下面的信息：

Total Internet sales in 2005: \$3,266,373.66

我们可以看到，因为销售量小于 330 万美元，字体颜色是红色的。

现在这个报表已经可以使用了。请一定要保存报表——这是我们经常做的操作。为了在 Report Builder 3.0 中查看报表，可以点击菜单栏的“Home”中的按钮“Run”。当报表出现时，输入一个年份(这里是从 2005 到 2008 之间的一个年份)，然后点击“View Report”。显示的报表应该类似图 15 所示的报表。



在本例中，销售总量接近于 980 万美元。正如我们所预期的，指针落在绿色范围内并且文本的字体颜色是绿色的。当然，实际的报表很可能比这个更复杂，外观更好，但是现在至少我们已经了解了如何添加一个仪表到一个报表中了。

随着你对仪表的使用越来越熟悉，你可能会尝试不同类型的仪表，并将它们用在不同的场合。例如，你可以将一个仪表放到一个模型、一个表中，或者使用多个仪表来比较数据。我在这里给你介绍的仅仅是简单的开头，但正如你所看到的，仪表能给我们一个很有价值的可视化工具，使我们的报表外观更加丰富和信息更加有效。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 使用 Report Builder 3.0 添加仪表完结篇

链接: http://www.searchdatabase.com.cn/showcontent_31163.htm

使用 Report Builder 3.0 添加图表

在本文中，我想介绍另一个重要的数据可视化工具——图表。如同仪表一样，图表能通过简明扼要的可视化格式显示有意义的数据来强化 Report Builder 3.0 报表效果。但是，图表能够描述远比仪表复杂的信息，并且图表可以有许多方式来保证数据的正确表示。

Report Builder 3.0 支持许多类型的图表，分别显示比例、线状或多值数据。下表列出了 Report Builder 3.0 所支持的 8 种图表类型和它们显示的数据类型：

Chart Type	Ratio	Linear	Multi-Value
Column		X	
Line	X		
Shape		X	
Bar		X	
Area		X	
Range		X	X
Scatter	X	X	
Polar	X		

每一种图表类型都支持许多本质上相似的图表。一般来说，我们可以根据需要切换图表类型(这个过程会随我们对图表的熟悉而变得简单)。对于每一种图表类型和它们所支持的详细图表信息，请查看 Report Builder 3.0 Help，这个帮助文件是与 Report Builder 一起安装的。

创建数据源和数据集

为了演示本文所使用的例子，我安装了一个本地的 [November 2009 CTP 版本](#) 的 SQL Server 2008 R2 实例。我还安装了 SQL Server 2008 R2 版本带的 [AdventureWorks 示例数据库](#)。除了 SQL Server，我还在同一个服务器上安装了 November CTP 版本的 Report Builder 3.0。

在我安装了 SQL Server 和 Report Builder 后，我新建了一个报表，并创建了一个数据源连接到 SQL Server 2008 R2 本地实例的 AdventureWorksDW2008R2 示例数据库。我将数据源命名为 AdventureWorksDW2008R2。然后，我创建一个名为 InternetSales 的数据集。该数据集使用下面的查询语句来查询 AdventureWorksDW2008R2 数据源：

```
SELECT
    c.FirstName + ' ' + c.LastName FullName,
    p.EnglishProductName ProductName,
    t.SalesTerritoryRegion SalesRegion,
    t.SalesTerritoryCountry SalesCountry,
    t.SalesTerritoryGroup SalesGroup,
    d.MonthNumberOfYear MonthNumber,
    d.EnglishMonthName OrderMonth,
    d.CalendarQuarter OrderQuarter,
    d.CalendarYear OrderYear,
    s.SalesAmount
FROM
    FactInternetSales s
    INNER JOIN DimCustomer c
    ON s.CustomerKey = c.CustomerKey
    INNER JOIN DimProduct p
    ON s.ProductKey = p.ProductKey
    INNER JOIN DimSalesTerritory t
    ON s.SalesTerritoryKey = t.SalesTerritoryKey
    INNER JOIN DimDate d
    ON s.OrderDateKey = d.DateKey
    WHERE
    d.CalendarYear = @OrderYear
```

注意 WHERE 子句将 CalendarYear 字段赋为变量@OrderYear 的值，这样用户可以在查看报表时指定一个年份。这个变量会作为参数自动地传递给报表。我们可以在 Dataset Properties 对话框的 Parameters 页或 Report Data 面板上的 Parameters 节点下方查看该参数。

这个数据集查询的数据库已经远远超过我们在图表示例中所需要的。然而，如果我们决定尝试这些例子，这些额外的数据对于我们了解更多图表高级特性会很有用，但是本文不会涉及这些高级特性。

注意：对于这些高级特性的详细信息，以及如何创建一个数据源和数据集，请参考 Report Builder 3.0 Help。

在我们创建了数据源和数据集后，我们可以添加图表到报表中了。要记住一个报表通常使用的特性并不仅限于图表，但在本文中，我们只关注于添加和配置图表。因此，这些图表会反映存储在 AdventureWorks 数据仓库中的 Internet 销售量。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 使用 Report Builder 3.0 添加图表

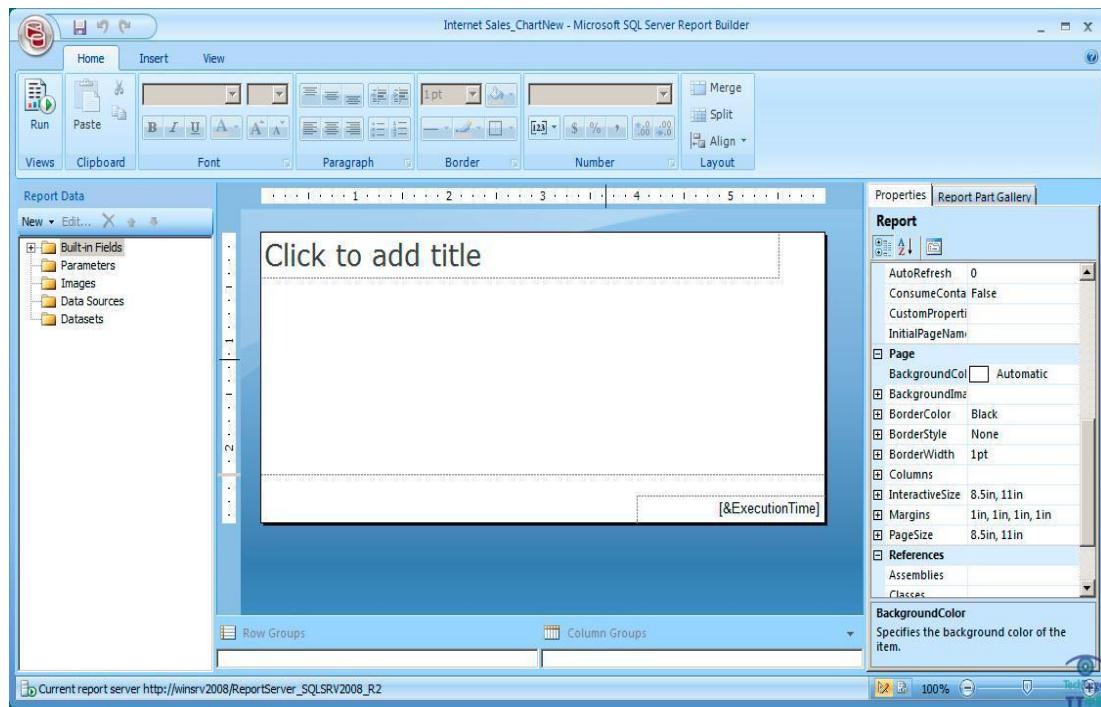
链接: http://www.searchdatabase.com.cn/showcontent_31524.htm

在 Report Builder 3.0 上创建条形图——第一部分

一旦准备好添加图表时，下一步就是确定使用哪一种类型的图表效果最佳了。第一部分详细说明了 Report Builder 3.0 中的柱状报表，并且概括了不同的柱状选项。

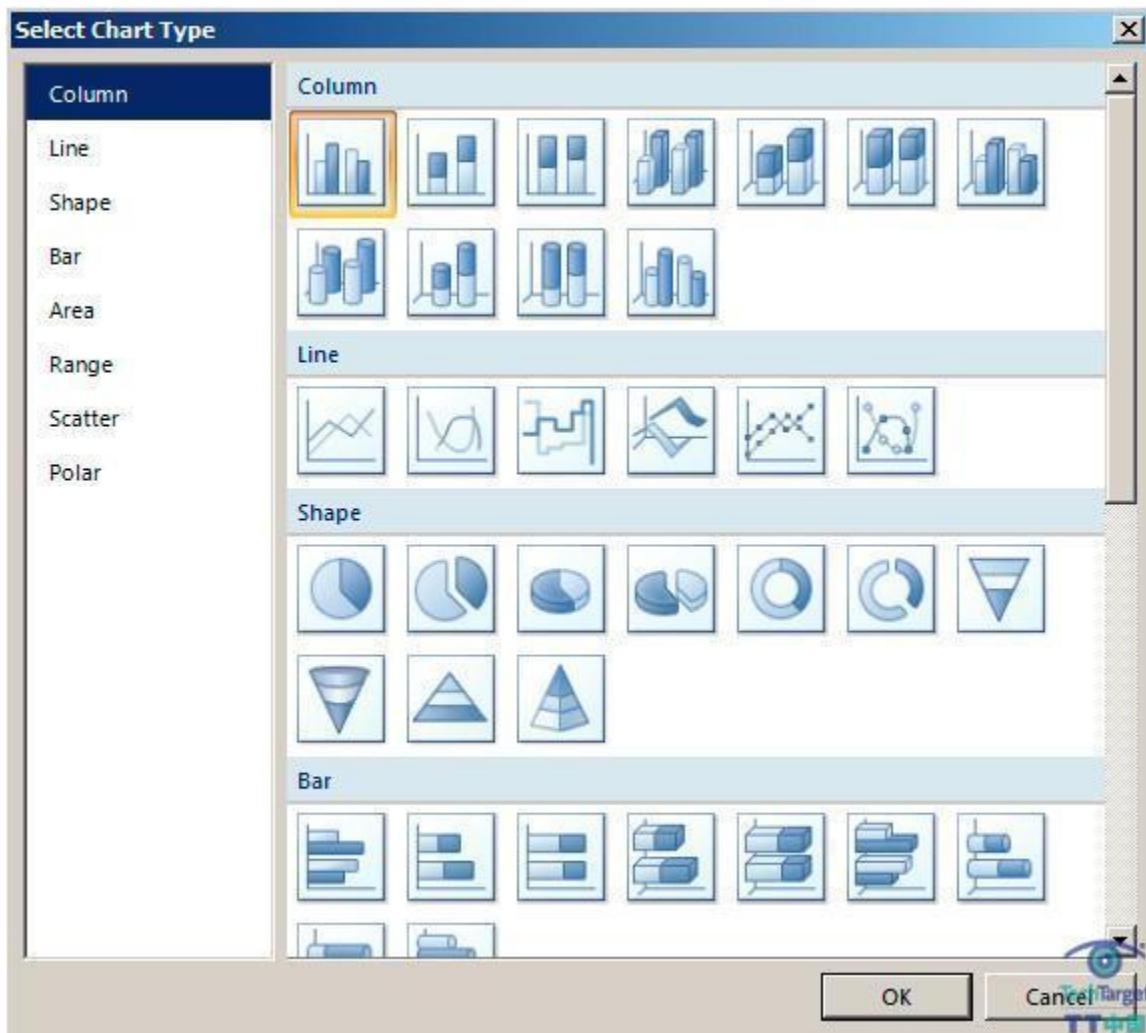
添加一个条形图到报表中

在新建一个报表时，我们会看到类似于图 1 的窗口设计板。在添加了数据源和数据集后，它们会被列在 Report Data 面板上，相应地位于 Data Sources 和 Datasets 节点下。如果数据集包括了查询变量，它们将会显示在 Parameters 节点下，这个我已经在前面的变量@OrderYear 说明中提到了。



选择菜单条的“Insert”，然后点击“Insert Chart”后的“Chart”，就可以添加图到报表中。这时当鼠标移到设计板上时，指针会变成迷你图表的图标。我们需要点击设计板，将虚线矩形拖到一个反映图表假定边界的形状上。当放开鼠标按钮时，Select Chart Type 对话窗口会出现，如图 2 所示。

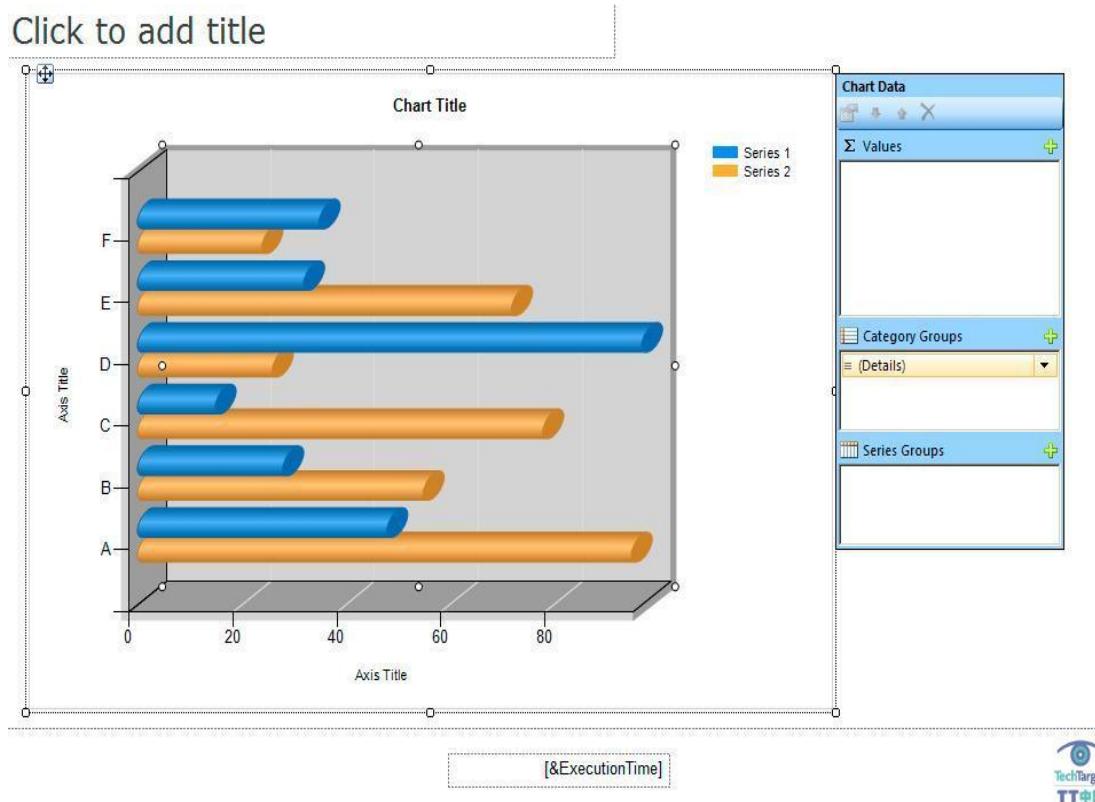
注意：Report Builder 3.0 也带有图表向导，它会指引我们完成创建图表的每一个步骤。然而，本文采用的是手工方法。



在 Select Chart Type 对话窗口中，我们可以从 8 个类别中选择所需要的图表。所选择的图表是可以在后面修改的。在 Report Builder 中切换图表类型是很简单的。

这里我想要演示的第一种图表是“条形图”。查看各种选项，我们可以点击图表类型的“Bar”。然后双击这部分的最后一个条形图——3-D Clustered Horizontal Cylinder。

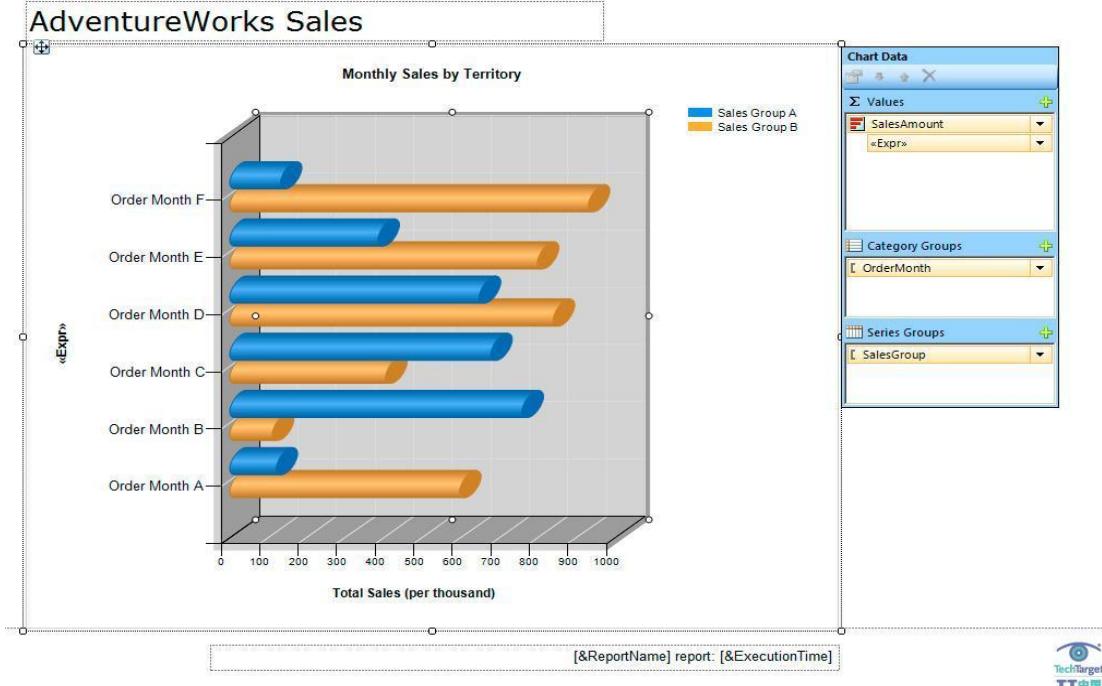
当双击图表时, Report Builder 会将它添加到我们之前用鼠标划定的区域, 如图 3 所示。注意图表右边的 Chart Data 面板。它只会在点击报表时才会显示。



图表数据是图表配置的最重要部分。它决定了图表上显示的信息及其显示方式。注意图 3 中 Chart Data 面板包括了 3 种类型的数据:

- 值——汇总显示在图表中的值。
- 类别组——数据片段的分组方式。
- 系列组——汇总数据的分组方式

对于大多数图表, 我们一般会从数据集中选择至少一个字段添加到每一个部分中。图 4 显示了我配置了属性的条形图。注意 Chart Data 面板中每一部分都包含了一个字段。



让我们依次查看每一部分的图表数据。在 Values 部分，我从数据集中选取了字段 SalesAmount。点击加号(+)按钮并选择该字段，就可以添加一个字段。当添加一个字段到 Values 部分时，它会自动地创建一个汇总数据的表达式。通常，该表达式会在字段下显示第二个列表，它一般是这样的：

Sum(Fields!SalesAmount.Value)

然而，为了说明条形图，我修改了基本表达式，这就是为什么现在表达式所在位置显示了如下占位符的原因：

< <Expr>>

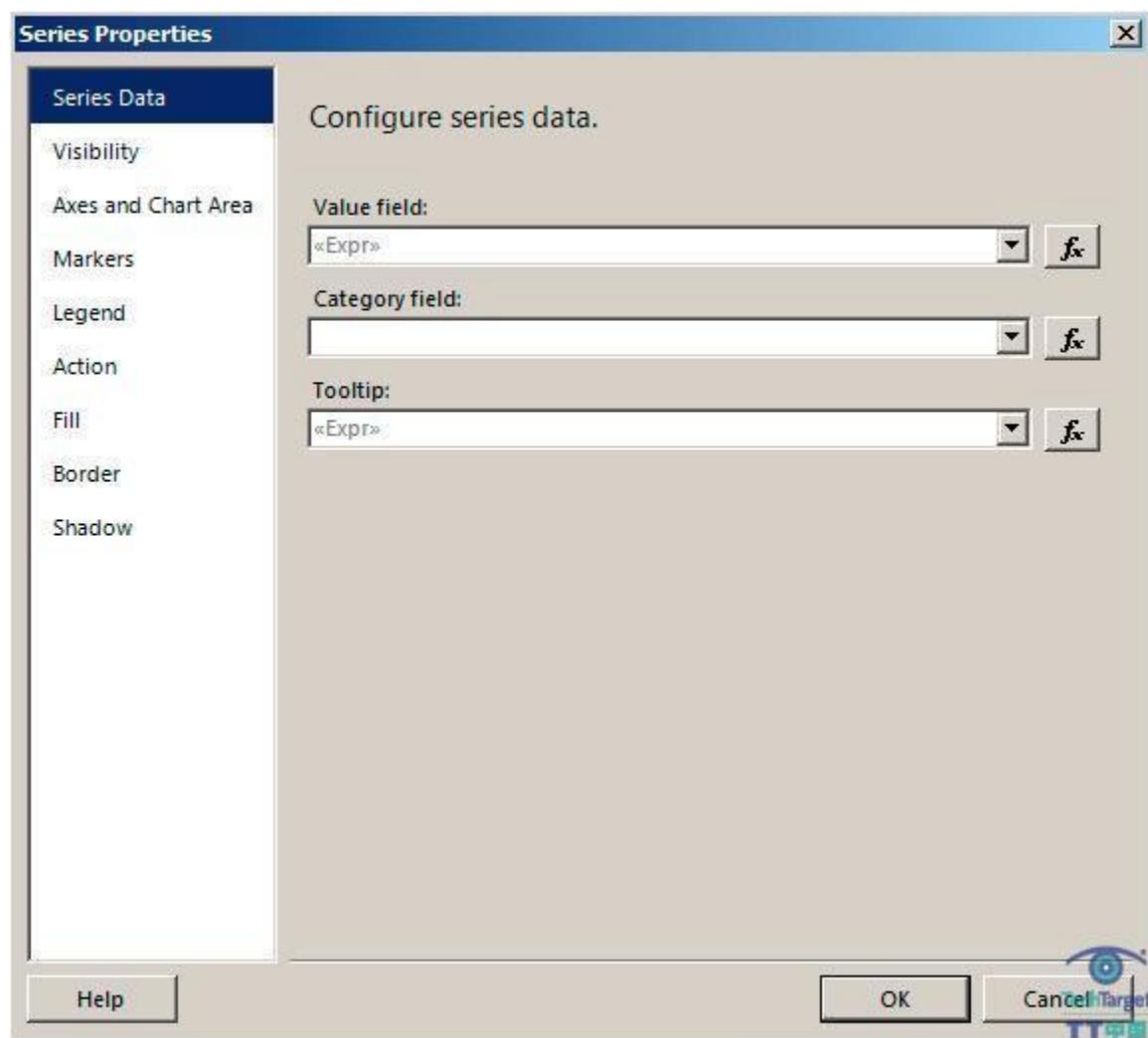
(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 在 Report Builder 3.0 上创建条形图——第一部分

链接: http://www.searchdatabase.com.cn/showcontent_31646.htm

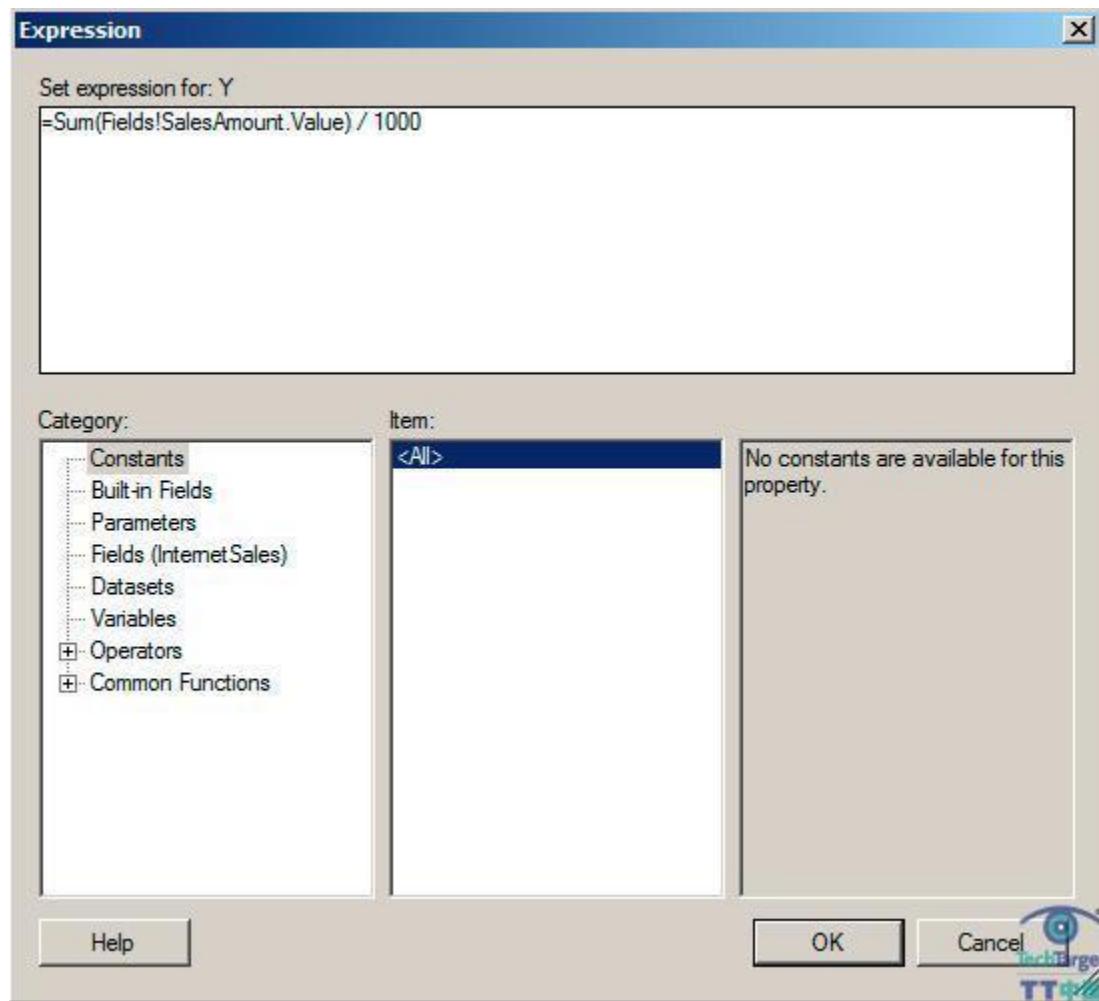
在 Report Builder 3.0 上创建条形图——第二部分

为了更好地理解表达式，让我们看一下 Values 数据的属性。右击字段名(这里是 SalesAmount)，然后点击“Series Properties” 打开 Series Properties 对话窗口就可以查看属性，如图 5 所示。



我们可以看到，Value 字段文本框显示了表达式占位符。点击表达式按钮(文本框右侧)，打开 Expression 对话窗口就可以查看该表达式。在对话框中，我们可以为图表中大多数属性创建表达式，包括系列值字段。对话框包含了作为参数、字段、变量、操作符和

方法的元素——所有这些都可以在表达式中使用。图 6 显示了我在条形图中为 SalesAmount 字段所创建的表达式。



对于这个表达式，我只是将汇总值除以 1,000，这样图表中数据会更容易读懂。这样，图表会以千为单位显示销售量，而不是显示所有数位。

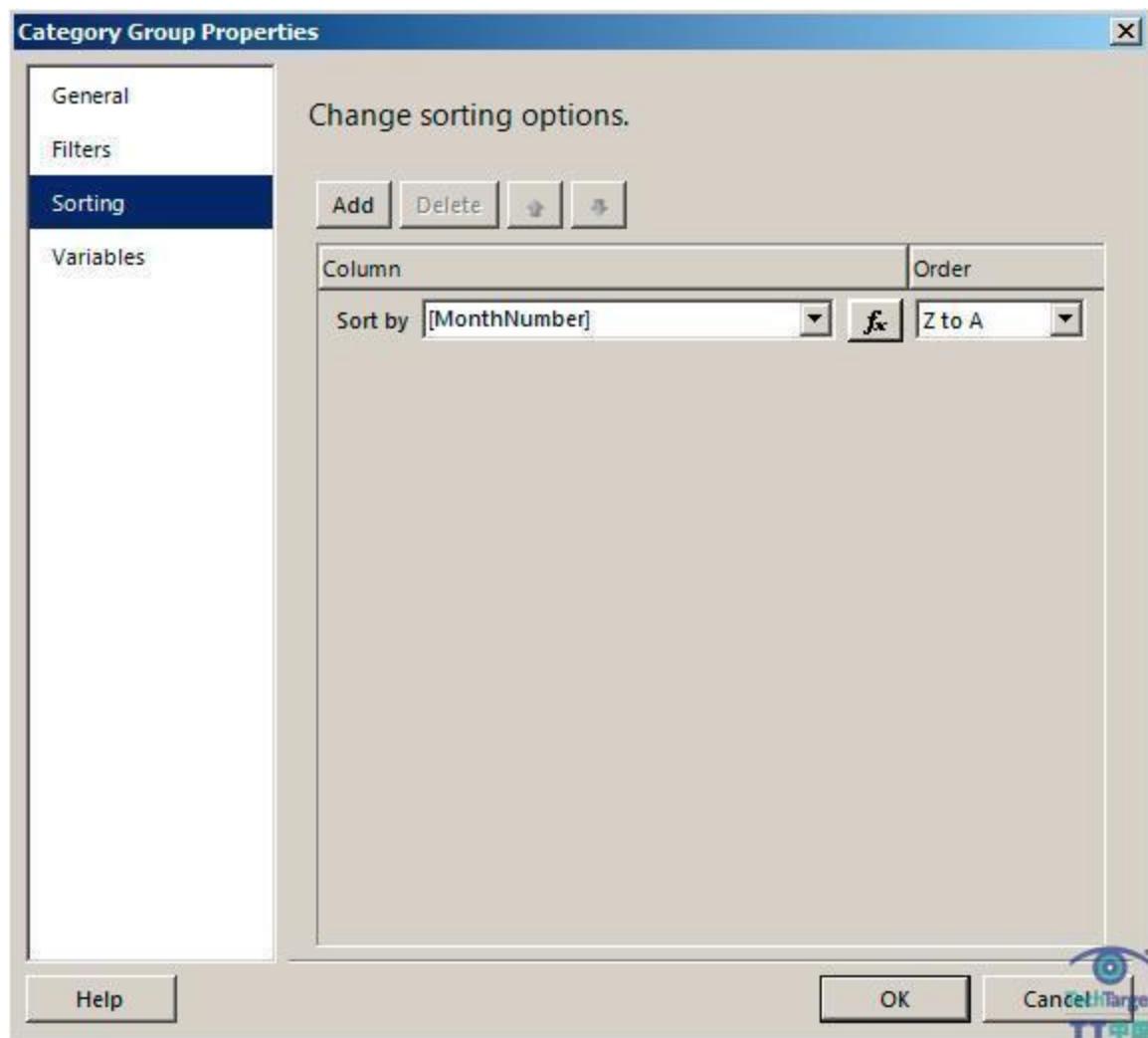
如果传到图 5，还可以看到我为 Tooltip 属性创建了一个表达式。这个属性允许我们在用户鼠标指向图表某个数据点时显示该数据。我使用了下面的表达，这样用户就能够看到确切的销售量：

```
=FormatCurrency(Sum(Fields!SalesAmount.Value))
```

对于这个表达式，我将汇总的销售量格式化成一个货币数，这样工具提示信息会更具可读性。

我还没有为条形图配置其它的系列属性，所以让我们转到 Chart Data 面板（如图 4 所示）的 Category Groups 部分。在这里，我添加了 OrderMonth 字段，它显示在图表的纵轴上。注意现在该数轴上每个节点的示例数据就是“Order Month”。

类似系列属性，我们可以修改分类组属性。右击 OrderMonth 字段，然后点击“Category Group Properties”打开 Category Group Properties 对话窗口，就可以看到分类组属性。我只修改了与排序相关的属性，如图 7 所示。



我先修改 Column 属性，这样它会引用 MonthNumber 字段，而不是 OrderMonth。否则，查询值会基于字母顺序排序，而不是按日期排序。此外，我还修改了 Order 属性。默认情况下字段是升序排列的(从 A 到 Z)。然而，因为月份是显示在纵轴上的，默认的排列顺序会从下到上列出所有月份。我将 Order 属性修改成“Z to A”以倒转顺序。

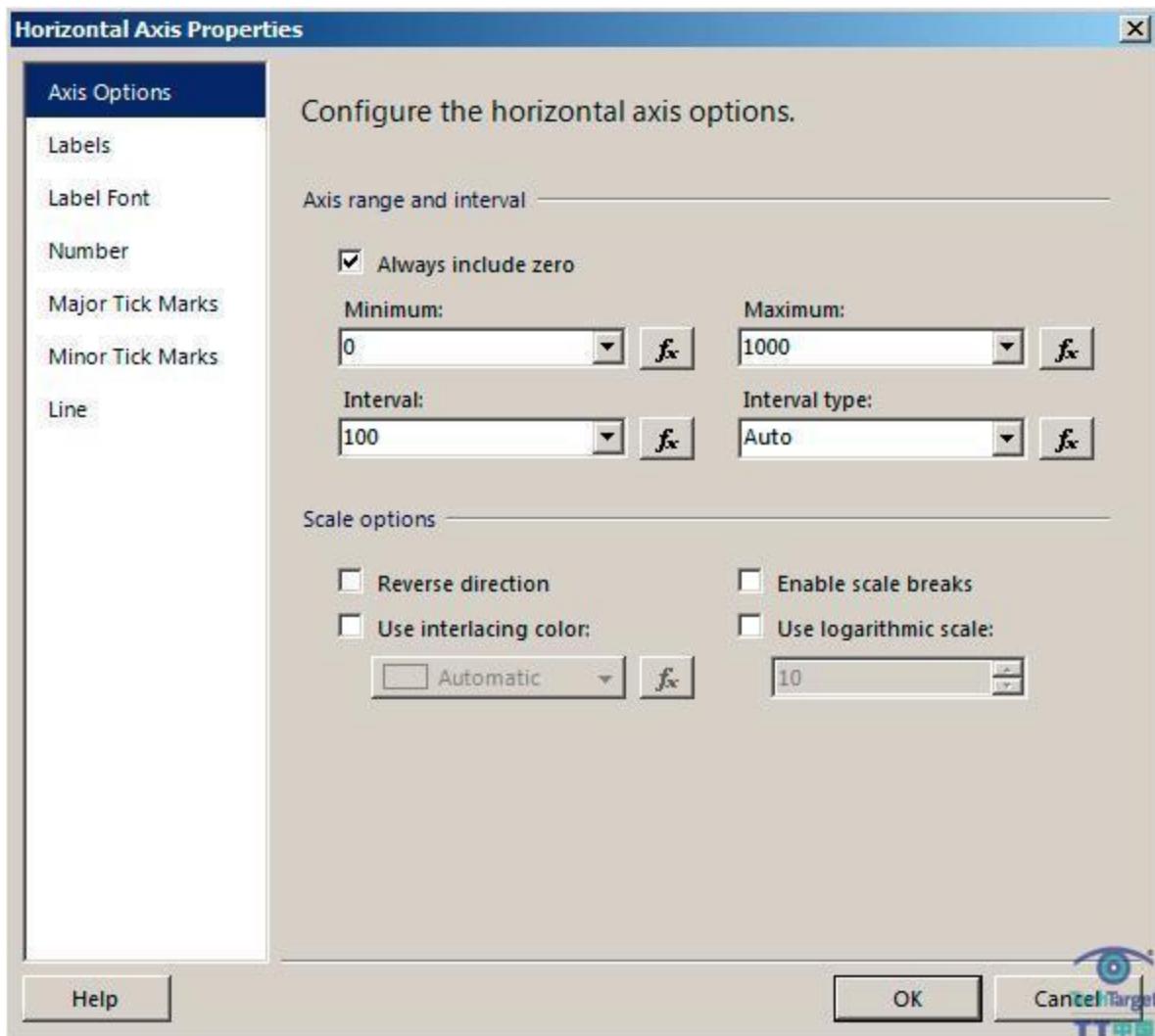
Chart Data 面板的最后一部分是 Series Groups。在这个示例柱状报表中，我添加了 SalesGroup 字段，它显示在示例数据图表的键上。我没有修改任何系列组属性。当数据显示在图表上时，它会基于这个字段的值进行分类。

这就是关于图表数据的所有方面。我之前已经提到，这是所需要配置的最重要属性。但是，仍然还有其它的属性对于图表也很重要。例如，我还将页脚信息修改成如下格式：

```
[&ReportName] report: [&ExecutionTime]
```

除了页脚默认包含的内置参数 ExecutionTime，我增加了内置参数 ReportName 和一些文本。我们可以在 Report Data 面板的 Built-in Fields 节点下查看内置参数列表。

我还修改了横轴属性。右击数轴，然后点击“Horizontal Axis Properties”打开 Horizontal Axis Properties 对话窗口，就可以修改横轴属性，如图 8 所示。你应该还记得，我之前候了系列字段的表达式使它显示以千为单位的销售量。这里我想让横轴也显示这些数值，所以我设置 Minimum 属性为 0，Maximum 属性为 100，Interval 属性为 100。其中最大值我选择 100 是因为我熟悉这些数据，并且知道总数不会超过 100,000。



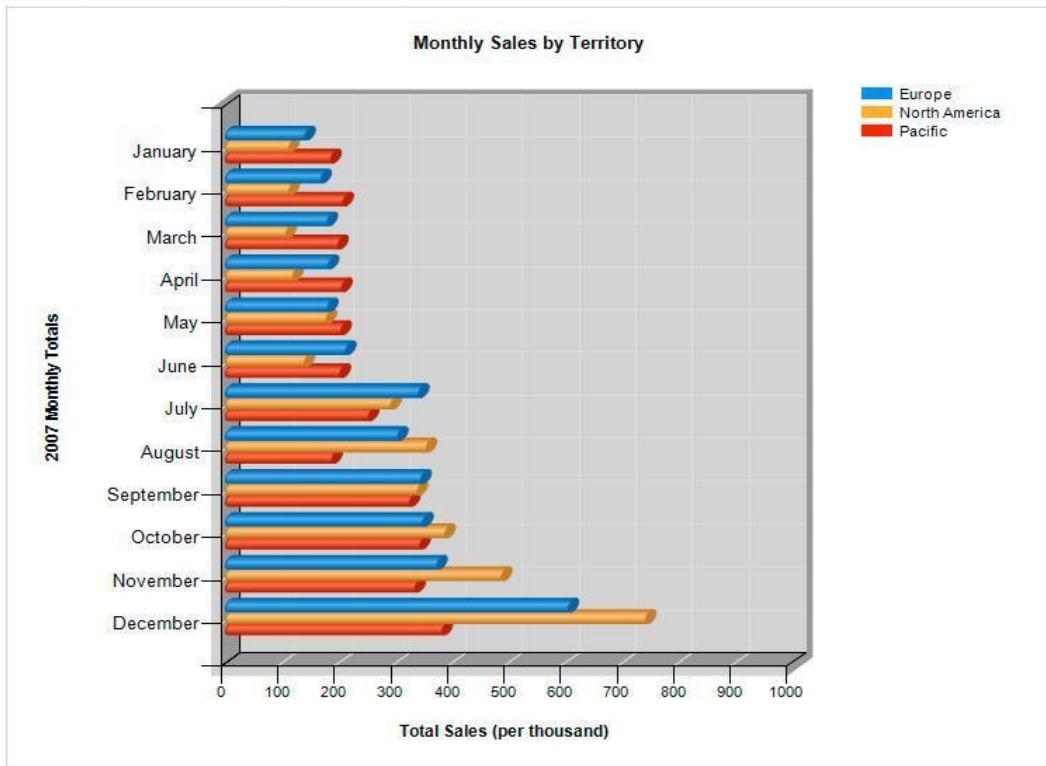
除了修改横轴属性，我还修改了纵轴属性的 Interval 属性为 1。我这样做是因为我想保证全年 12 个月份都在报表中显示。否则，Report Builder 3.0 只会显示一部分的值，如隔月显示。

回顾一下图 4，可以从表达式占位符看出我也修改了纵轴的数轴标题。右击数轴标题文本框，然后点击“Axis Title Properties”，就可以查看数轴标题。这里，我们可以修改文本或填写表达式。我添加了如下的表达式：

```
=Parameters!OrderYear.Value + " Monthly Totals"
```

这个表达式从参数 OrderYear 查询值，然后在报表渲染时显示该值。不管用户输入什么年份，它都会显示在数轴标题上，如图 9 所示。注意，图表的横轴也显示 0 到 1,000 间隔为 100 的范围。此外，月份在纵轴从上到下列出，图表键显示销售组的名称：Europe、North America 和 Pacific。

AdventureWorks Sales



InternetSales_BarChart report: 12/29/2009 6:14:21 PM


如果创建了我这里所介绍的条形图，鼠标悬停在每个列上可以看到每个汇总值，它们会显示在工具提示上。此外，我们还可以通过输入不同的年份作为参数，然后重新生成报表，就可以显示每年的数据结果。

虽然我这里介绍的属性配置是应用到示例条形图的，但是这些概念也是可以应用到其它图表上的。所以，让我们看一些其它类型的图表，以帮助你了解它们的工作方式。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 在 Report Builder 3.0 上创建条形图——第二部分

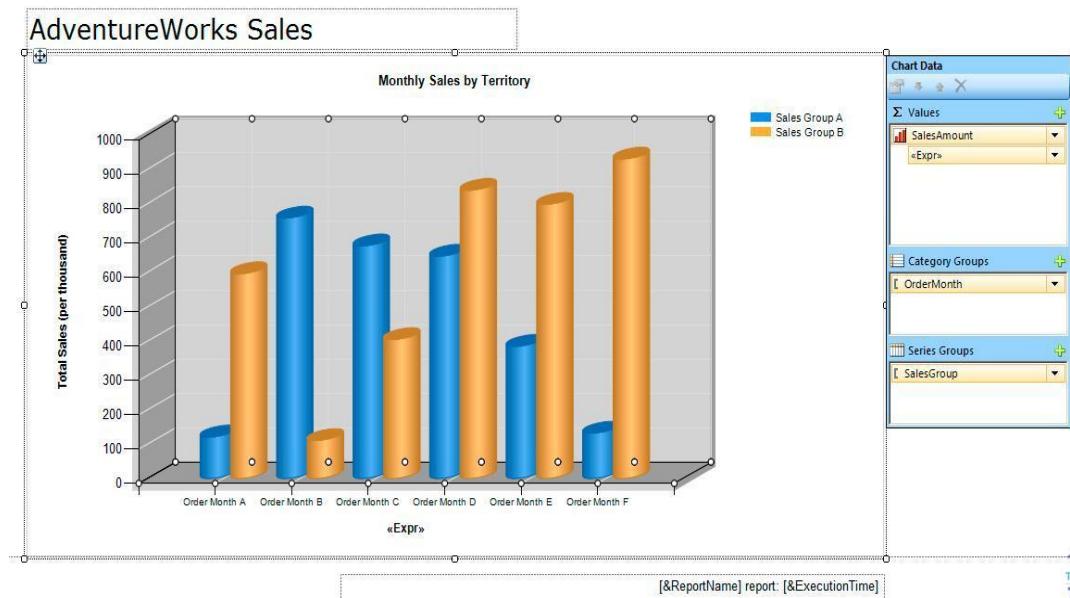
链接: http://www.searchdatabase.com.cn/showcontent_31647.htm

在 Report Builder 3.0 中生成柱状图和折线图

[第一部分](#)和[第二部分](#)阐述了在 Report Builder 3.0 中创建图表的过程——特别是强化数据可视化效果的条形图。另外 2 种常用的图表类型是柱状图和折线图。第三部分将阐述如何在 SSRS 报表中创建这 2 种图表。

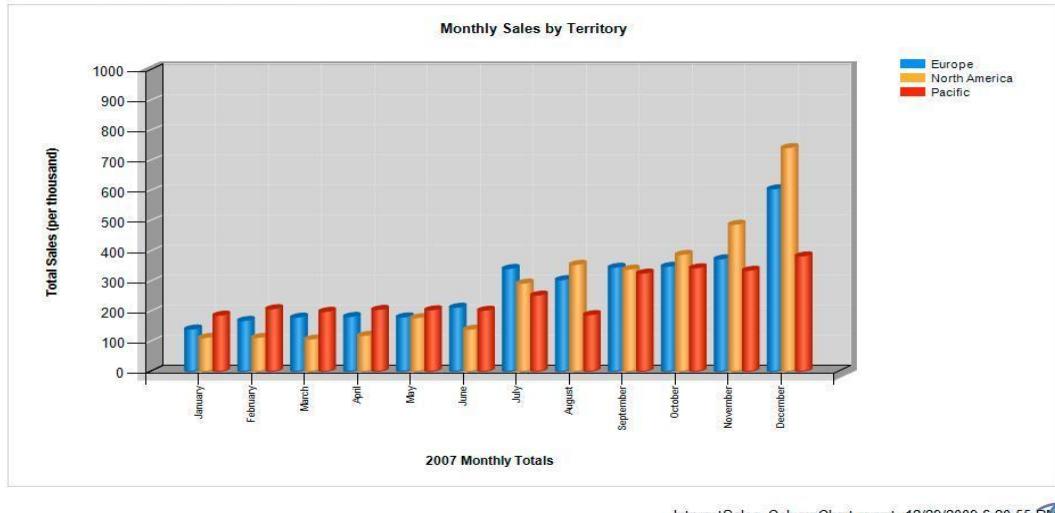
添加柱状图到报表中

柱状图类似于条形图，只是纵轴与横纵相对调。我们可以在设计视图中右击条形图，然后点击“Change Chart Type”，将条形图转换成柱状图。然后，我选择 3-D Clustered Cylinder 柱状图，如图 10 所示。



对于许多图表，我们可以不需要作太多修改就直接从一种类型转换成另一种。在这里，我所要做的唯一修改是调整图表的位置，使之在新样式下显示更合理些；并且，我也将分类组的排列顺序改回到默认顺序（A 到 Z）。现在渲染的报表显示为新的图表类型，如图 11 所示。

AdventureWorks Sales

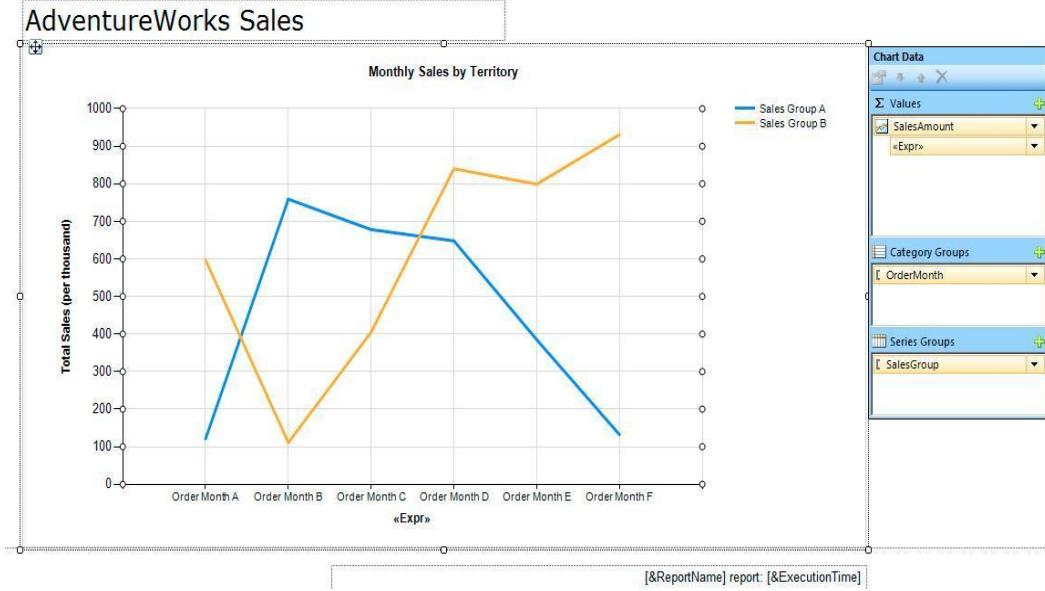


InternetSales_ColumnChart report: 12/29/2009 6:20:55 PM

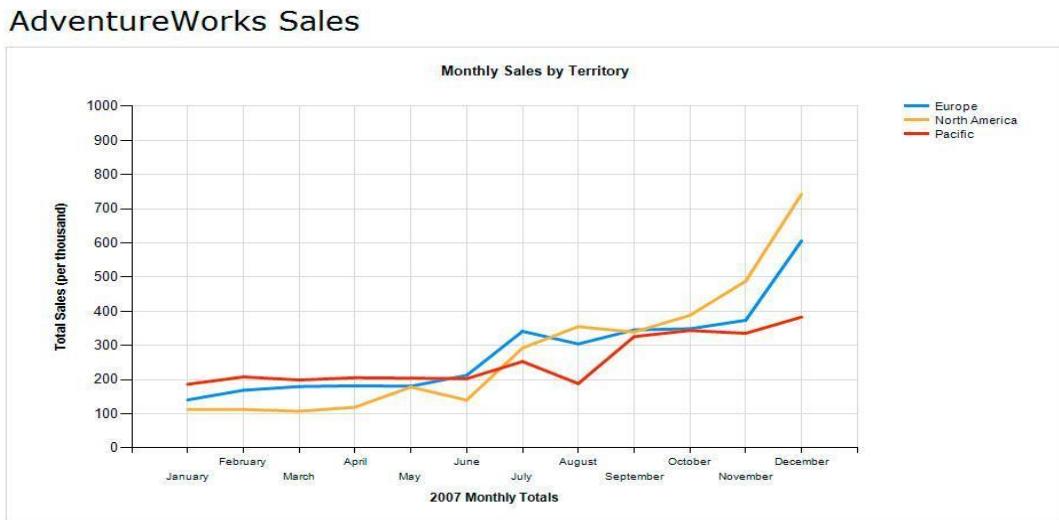

注意柱状图会比条形图更宽和矮一些，同时月份是从左到右以字母顺序显示的。然而，图表的大多数部分都保持原先条形图相同的属性设置。

添加折线图到报表中

将图表转换成折线图表也一样简单。图 12 显示的是我从柱状图表转换而来的一个基本的折线图表。其中唯一的修改是系列属性的 Line Width 属性。我将宽度修改为 2 像素（默认是 1 像素）。Line Width 属性可以在 Series Properties 对话窗口的 Border 页设置。



虽然从一个柱状图或条形图转换到一个折线图非常简单，但是图表渲染方式的差别是很显然的。图 13 显示了 2007 年的报表，它被渲染成一个折线图。



对于我来说，这个报表看起来很整齐、简单易懂。它包含了所有相同的信息，我们也可以将鼠标停在数据点查看当月的具体汇总值。当然，选择图表是一个很主观的决定，但是重要的是要记住我们的目标是用尽可能简单的方法表示最简明的信息，这样用户才能快速地获得他们需要的信息。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 在 Report Builder 3.0 中生成柱状图和折线图

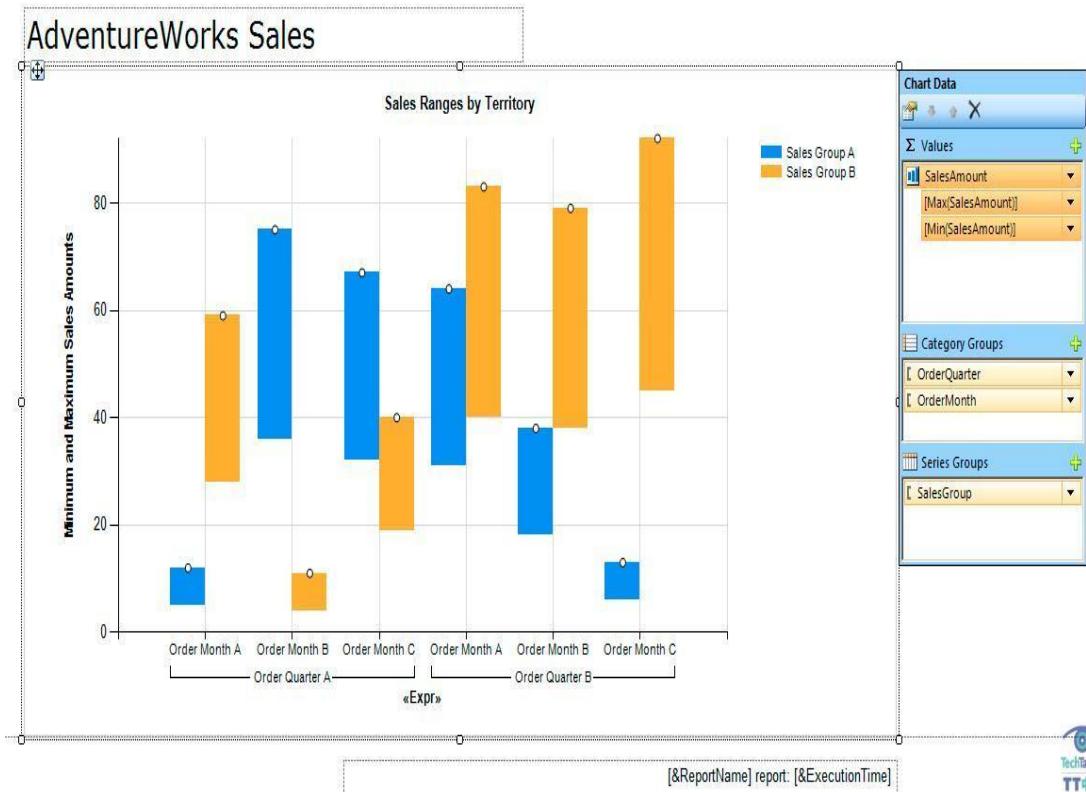
链接: http://www.searchdatabase.com.cn/showcontent_31677.htm

在 Report Builder 3.0 中使用区间图

本文将阐述区间图在 SQL Server Reporting Services 中的适用范围，以及它们的创建方法。

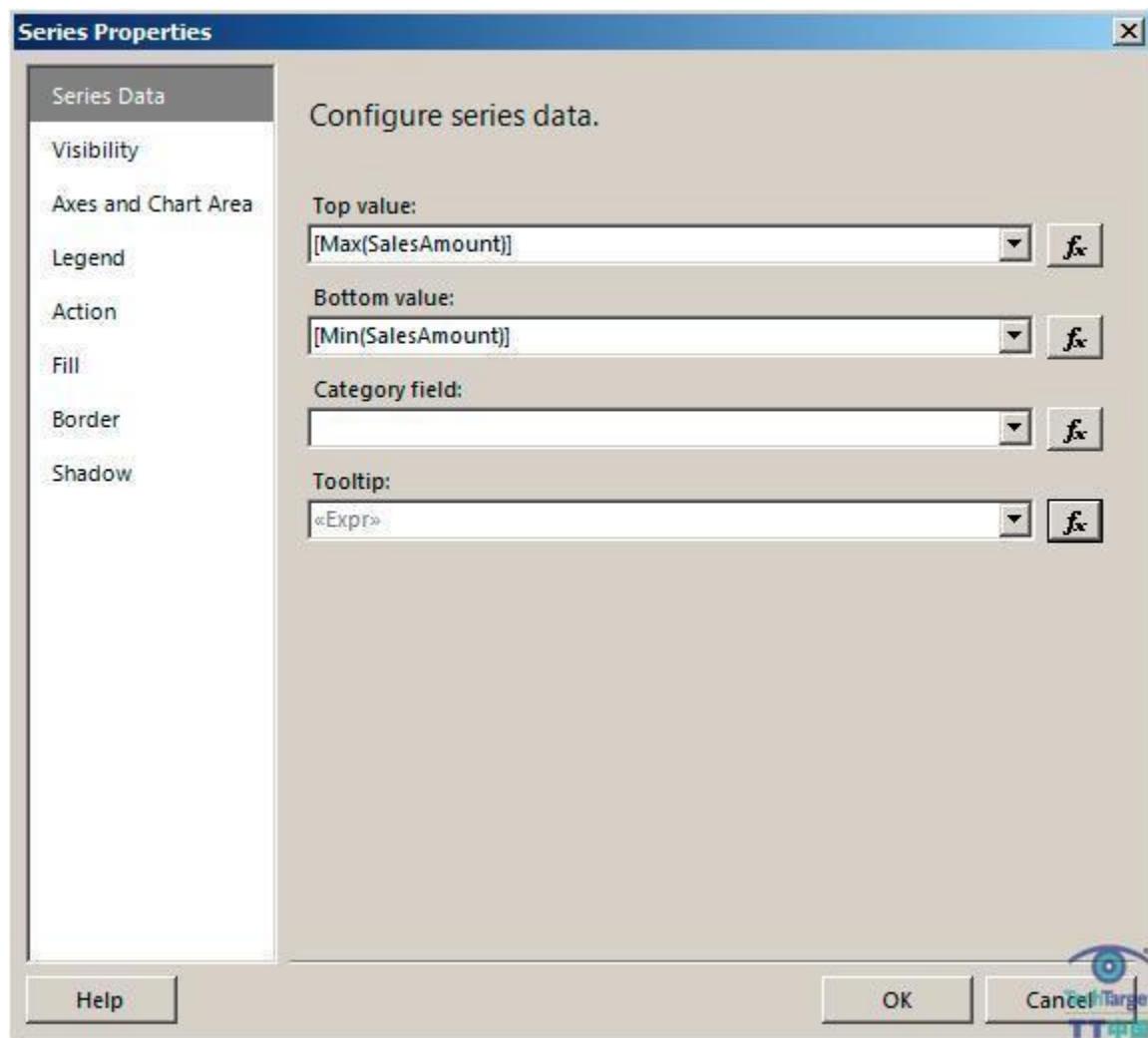
添加区间图到报表中

现在让我看一种稍微不同的图表——区间图。这种图表将基于 Chart Data 面板的 Values 部分所包含的字段显示数据范围。例如，图 14 显示的是一个跟踪每个月的最小和最大销售额的区间图。



对于每个月，图表将会确定最高和最低的销售额，然后在图表上为每组数据点绘一个条形。图 15 显示了 Series Properties 对话窗口中的 Top 和 Bottom 值属性。注意上限值

使用 Max 函数，而下限值使用 Min 函数。当然，这与在大多数报表中使用的默认的 Sum 函数是不同的。



除了值相关的域，我也在 Tooltip 属性中使用与之前图表不同的表达式：

```
= "Min/" & FormatCurrency(Min(Fields!SalesAmount.Value)) & " " & "Max/" &  
FormatCurrency(Max(Fields!SalesAmount.Value))
```

现在当用户鼠标停在其中一个列时，工具提示将显示最小值和最大值。

另一个与之前报表设置不同的系列属性是 Color 属性(在 Series Properties 对话窗口的 Fill 页面上)。这里我没有使用默认值，而是通过定义下面的表达式来设置颜色：

```
=Iif(Fields!SalesGroup.Value = "Pacific", "Aqua",
Iif(Fields!SalesGroup.Value = "Europe", "Yellow", "Blue"))
```

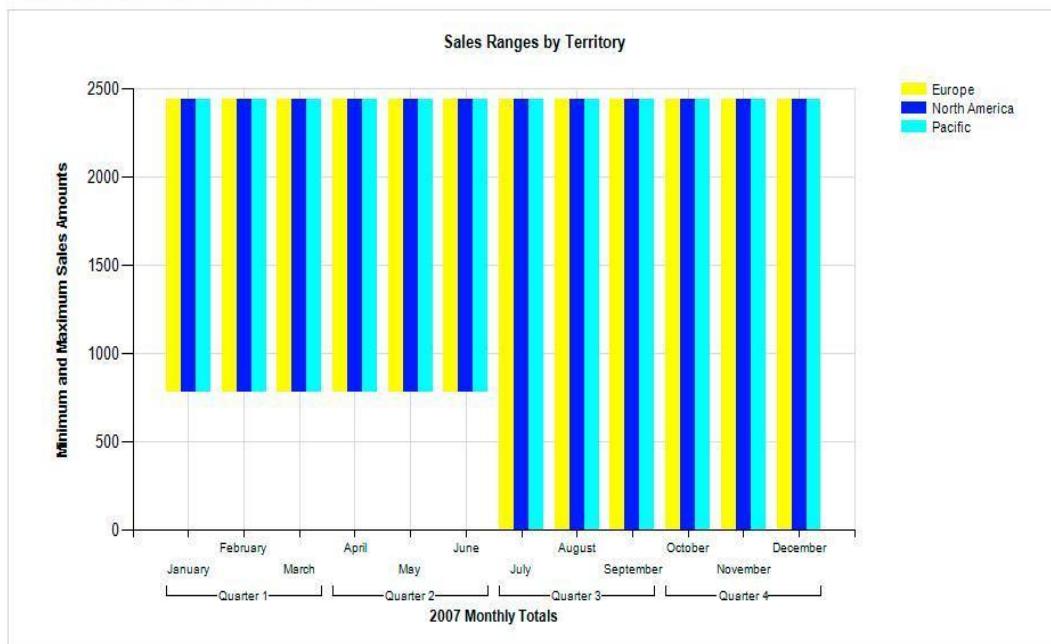
如果销售组是 Pacific, 那么该列将会是显示成浅绿色。如果该组是 Europe, 显示颜色将会是黄色。否则, 颜色会是蓝色。

另一个与之前图表不同的是分类组。对于区间图, 我指定了 2 个分类组: OrderQuarter 和 OrderMonth。这样, 横轴会显示一个额外的子分类, 虽然这不会改变显示的列和值。同时, 因为 OrderQuarter 值是存储为 1、2、3 和 4, 我将 Label 属性修改成以下的表达式:

```
="Quarter " & Fields!OrderQuarter.Value
```

在这个表达式中, 我只是简单地拼接文本和 OrderQuarter 字段值。现在标签文本将会包含单词“Quarter”。图 16 显示的是用户查看 2007 年报表时显示的图表。

AdventureWorks Sales



InternetSales_RangeChart report: 12/29/2009 6:34:15 PM

TechTarget | 中国

我们可以看到，每个月显示了 3 列反映最小和最大销售额的区间。这个图表反映了我设置的颜色，而季度则列在横轴下文。(注意 AdventureWorks 数据仓库的示例数据并不是最好的，但是它仍然足够让我们了解图表的工作方式。)

(作者: *Robert Sheldon* 译者: 曾少宁 来源: TT 中国)

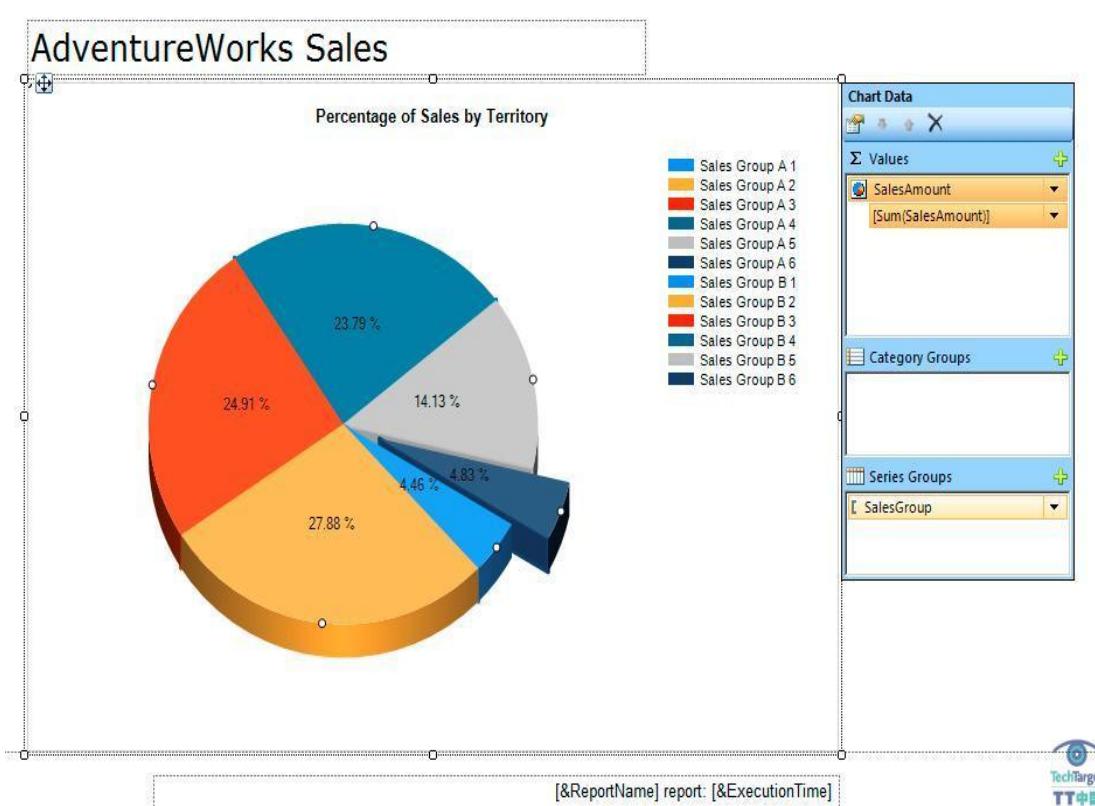
原文标题: 在 Report Builder 3.0 中使用区间图

链接: http://www.searchdatabase.com.cn/showcontent_31806.htm

在 Report Builder 3.0 中使用饼状图

添加饼状图到报表中

饼状图比之前我们所知道的图都要简单。事实上，它就应该是很简单的。因为形状图的目的是显示数据分组之间的简单对比(通常是百分比)。如果这些图表太复杂，就失去它们本来的作用了。图 17 显示了我添加到报表中的饼状图。其中这是一种 3-D Exploded Pie 图(立体饼状图)。



首先，我们注意到系列值域 SalesAmount 只是使用 Sum 函数来汇总数据。对于数字域这是默认的汇总方法，所以当添加域到图表数据时，Report Builder 3.0 会自动地使用 Sum 函数来进行汇总。此外，我在系列组中包含了 SalesGroup 域，但是我没有在分类组中添加任何域。这个图表将会简单地显示某个年份每个组的销售额所占百分比。

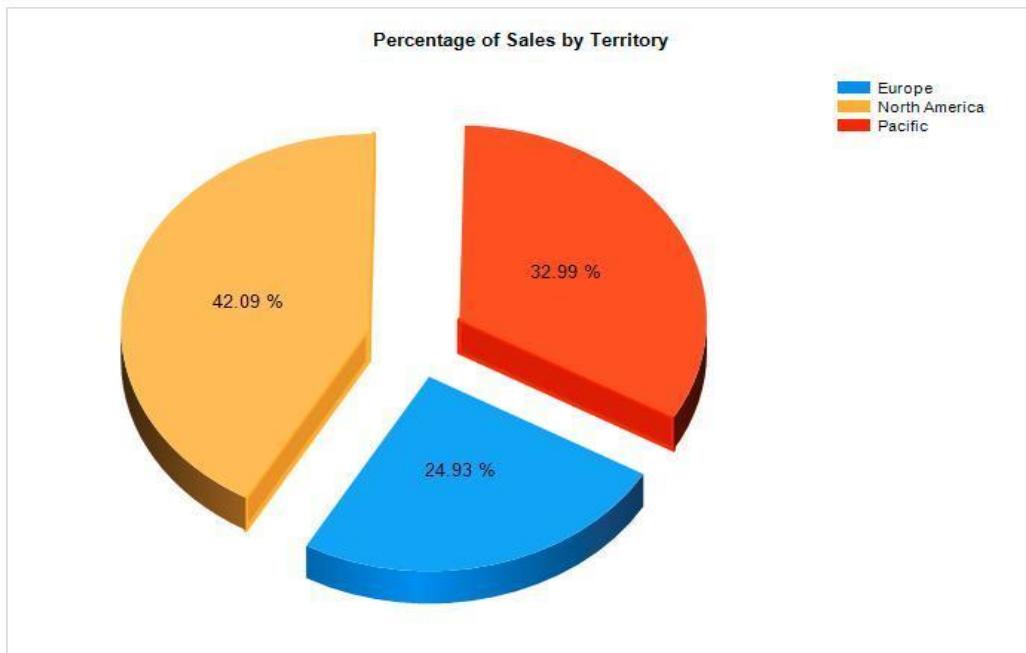
在我添加了域到图表数据后，我对图表所作的唯一重要修改是增加了标签到饼的每个部分。右击图表，然后选择“Show Data Labels”，就可以添加标签。虽然默认情况下标签不会显示百分号，但是我们可以通过配置标签属性来使它显示百分号。

右击饼状图中的其中一个标签，然后点击“Series Label Properties”，就可以修改属性。在 Series Label Properties 对话窗口的 General 选项卡中，在 Label Data 文本框中输入以下的表达式：

#PERCENT

这就是创建一个饼状图的所有步骤。当图表渲染时，它会将饼分成几个部分，每一个部分对应一个销售区域，如图 18 所示。

AdventureWorks Sales



InternetSales_PieChart report: 12/29/2009 6:38:48 PM
 TechTarget
TT 中国

注意每一个部分都显示了一个百分值。我们可以看到，如果添加太多的信息到饼状图，它会很快就会变得很混乱。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

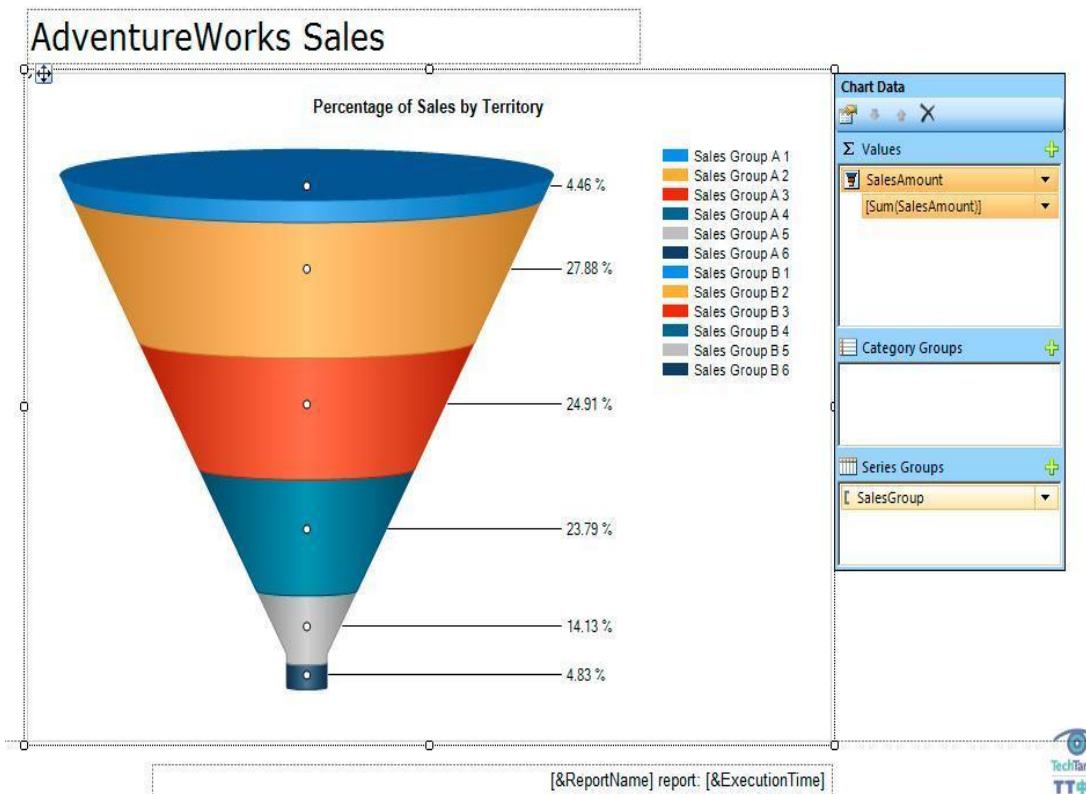
原文标题: 在 Report Builder 3.0 中使用饼状图

链接: http://www.searchdatabase.com.cn/showcontent_31807.htm

在 Report Builder 3.0 中使用漏斗图

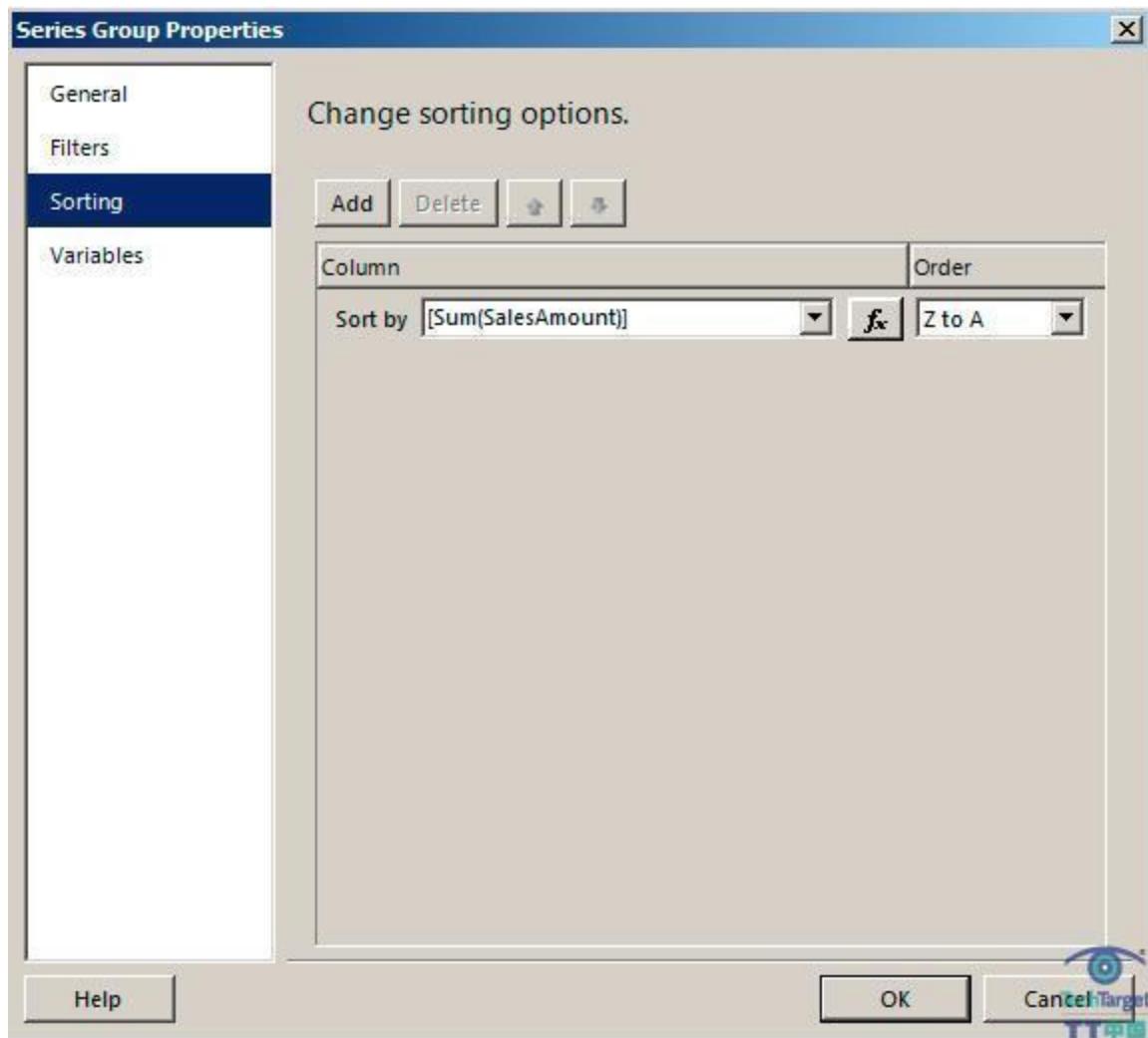
添加漏斗图到报表中

另一种形状图表是漏斗图。我们可以简单地从一个饼状图转换成一个漏斗图，而不需要修改任何配置。图 19 显示的是我从饼状图转换而来的漏斗图。



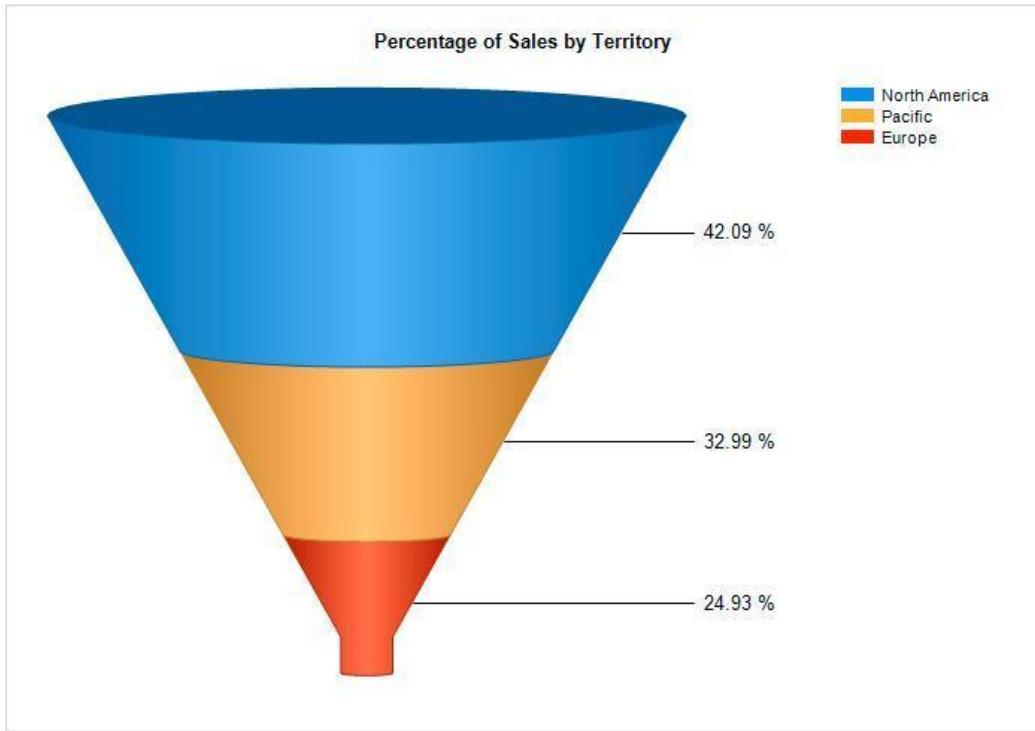
我们可以看到，图表数据看起来与饼状图差不多。然而，其中我作了一处修改，它可以改善漏斗图的显示效果。默认情况下，漏斗图的每个部分是根据 SalesGroup 值排序的，从 A 到 Z。对于饼状图，这没有什么分别。但是对于漏斗图，最好是根据汇总的系列值 SalesAmount 对各部分进行排序。

为了根据销售额排序漏斗各部分，我修改了系列组属性，如图 20 所示。



注意，现在数据是根据汇总的销售额从高到低(Z 到 A)进行排序。这样，销售最高的组在漏斗的顶部，而最少的组则在底部，如图 21 所示。

AdventureWorks Sales



InternetSales_FunnelChart report: 12/29/2009 6:41:07 PM
TechTarget
TT 中国

显然，创建形状图是很容易的，同时形状图能够快速地表示基本的重要统计信息。当然，对于更复杂的数据我们需要选择不同的图表。

Report Builder 3.0 支持许多类型的图表，远不止我这里所介绍的。对于每一种图表，都有一类属性来灵活配置图表在报表渲染。当然，我只是介绍了图表使用的基本方法。即使如此，这里的例子也帮助开始学习如何向报表中添加图表。当你准备使用图表时，记住一点就是图表可以很好地帮你解决报表优化的问题。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 在 Report Builder 3.0 中使用漏斗图

链接: http://www.searchdatabase.com.cn/showcontent_31808.htm

SQL Server 2008 R2：使地理绘图更简单

Report Builder 3.0 的其中一个最令人兴奋的就是新的地图功能，它可以添加由 1 个或多个显示地理和分析数据的层组成的地图。这个地图也可以与 Microsoft Virtual Earth 整合，以实现一个生动的数据可视化工具。

虽然我们可以创建一个只包含地理数据的地图，但是最有效的地图需要同时包含分析数据。例如，我们想要创建一个显示美国一个州每个县的总销售额。其中销售量代表分析数据，而绘制的县位置代表地理数据。

一个 Report Builder 地图是由 1 个或多个层组成的。每一层可以配置为下面的其中一种类型：

- 多边形——显示区域轮廓，如城市，州或国家。
- 点——显示地图上具体的位置点。
- 线——显示点之间的路径或路线。
- 块——由 [Microsoft Virtual Earth](#) (Bing) 地图构成的背景。

当创建一个地图时，我们需要通过一次添加一个层来实现。然后我们可以根据具体的需要配置每一个层。同时，其中配置的选择会根据每一层不同的类型而有所差别，即使其中有许多相似的地方。

在本系列文章中，我将介绍如何使用 Report Builder 3.0 创建一个包含多边形层、点层和块层的地图。这个图将显示美国陆地和销售客超过 1 百万美元的地方的销售人员。除了这 3 层之外，该地图还将包含显示每个人的销售额及销售对比所需要用到的图例、工具提示和颜色标注。

创建数据源和数据集

为了说明这个地图的创建方法，我首先创建一个报表并添加一个数据源和数据集。这个数据源连接到 SQL Server 2008 R2 的一个本地实例的示例数据库 AdventureWorks2008R2。我将该数据源命令为与数据库相同的名称。

我在一台 Windows Server 2008 主机上安装了 [November 2009 CTP 版本](#) 的 SQL Server 2008 R2 本地实例。这个安装包括了数据库引擎和 SQL Server Reporting Services (SSRS)。我还使用 SQL Server 2008 R2 版本所带的 [AdventureWorks 示例数据库](#)，并在同一台服务器上安装了 November CTP 版本的 [Report Builder 3.0](#)。

在我创建了数据源后，我创建了一个名为 SalesPeople 的数据集。该数据集使用下面的 Transact-SQL 语句来查询 AdventureWorks2008R2 数据源的数据：

```
SELECT
    (p.FirstName + ' ' + p.LastName) AS FullName,
    a.City,
    RTRIM(st.StateProvinceCode) AS StateProvinceCode,
    sp.SalesYTD,
    a.SpatialLocation
    FROM
        Sales.SalesPerson sp
    INNER JOIN Person.Person p
    ON sp.BusinessEntityID = p.BusinessEntityID
    INNER JOIN Person.BusinessEntityAddress bea
    ON bea.BusinessEntityID = p.BusinessEntityID
    INNER JOIN Person.Address a
    ON a.AddressID = bea.AddressID
    INNER JOIN Person.StateProvince st
    ON st.StateProvinceID = a.StateProvinceID
    WHERE
        p.PersonType = 'SP'
        AND st.CountryRegionCode = 'US'
        AND SalesYTD > 1000000;
```

上面的查询获取一组美国的销售人员，他们的销售客都超过 1 百万美元。除了查询销售人员的姓名外，我还查询了城市、州、销售总额和每一个销售人员的地理位置。地理位置是从 SpatialLocation 字段查询的，这个字段配置的是地理数据类型，这是新的 SQL Server 2008 地理数据类型之一。同时，注意我使用了 RTRIM 方法来修剪 StateProvinceCode 字段。在后面，我将解释为什么需要去掉后面的空格。

在创建报表并建立数据源和数据集后，我们就可以添加地图到报表中。注意除了地图，报表中还有许多其它的元素，但是这里我只关注于地图。如果想要学习如何添加其它元素

到报表中，或者如何创建一个数据源或数据集，请参考 Report Builder 3.0 Help 或者我之前关于[仪表](#)和[图表](#)的文章。

(作者: *Robert Sheldon* 译者: 曾少宁 来源: TT 中国)

原文标题: SQL Server 2008 R2: 使地理绘图更简单

链接: http://www.searchdatabase.com.cn/showcontent_32083.htm

在 Report Builder 3.0 中添加一个多边形图层到地图

在报表创建好后，我们可以开始添加包含空间和分析数据的地图了。但是从哪里开始呢？第二部分会介绍在 Report Builder 3.0 中创建地图的第一个步骤——多边形图层。

我们可以通过创建第一个层来插入一个新地图到报表中，而创建层的最简单方法是使用“New Map”向导。点击“Insert”菜单上“Map”图标下文的“Map Wizard”启动向导。当向导启动后，第一个页面提示我们选择空间数据源。我们可以从以上 3 种数据源中选择一个：

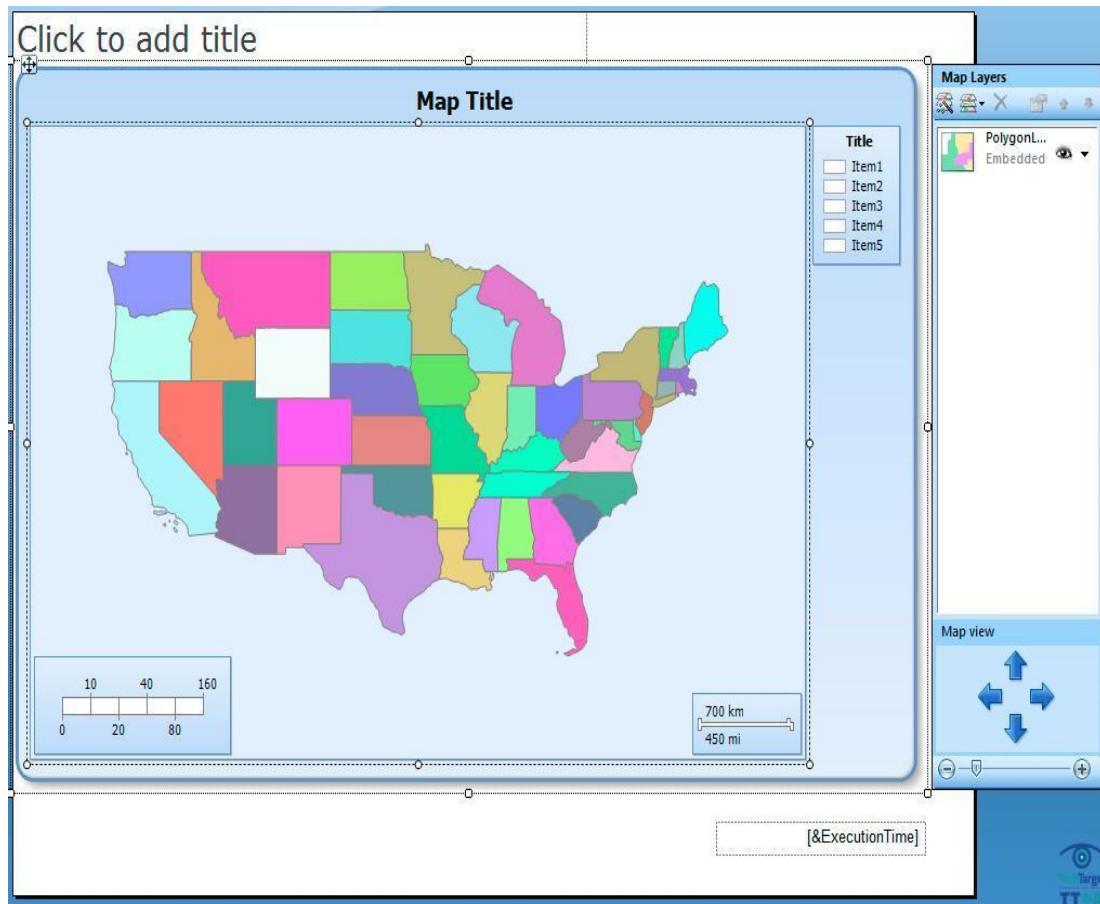
- Map Gallery——嵌入了空间数据的报表。Map Gallery 是在安装 Report Builder 3.0 时默认安装的。这个地图库包括了美国和各个州的地图。
- ESRI Shapefile——包含符合由 Environmental Systems Research Institute (ESRI) 确定的 Shapefile 空间数据格式的数据。这些文件指定了地理和几何形状。
- SQL Server 地理查询——来自一个包含配置了几何或地理空间数据类型字段的 SQL Server 数据库的数据。

在我们的第一层上，我们将使用 Map Gallery。在使用 Map Gallery 时，有一个多边形图层会自动创建，同时我们不使用 Map Gallery 来创建其它的层。此外，当使用 Map Gallery 时，我们必须选择向导屏幕下半部分的 Map Gallery 树中的地图。在本例中，我们将使用国家地图中的美国地图。

New Map 向导的剩余部分，我们直接点击向导屏幕接受所有的默认设置。这样我们将创建一个基本的多边形图层并使用默认的视图和颜色选项。当向导完成后，第一个地图层将增加到设计面板中。然后我们可以重新设置合适的地图大小——不过要小心不要扭曲国家的形状。

默认，地图会包括所有 50 个州。然而，根据这个报表的要求，我们将只使用陆地的州，所以我们可以删除 Alaska 和 Hawaii 这两个州。我们可以删除一些州，因为这个地图实际上是由各个多边形组成的；一个多边形代表一个个州。右击一个多边形，然后选择

“Delete Polygon”就可以删除这个州。然后这个地图将自动地放大并居中显示剩下的州，如图 1 所示。



当删除这 2 个州后，Map Layers 面板会出现(如果没有显示，再双击一下地图)。Map Layers 面板显示了我们创建的每一个图层的详细信息。到现在为止，应该只列出了多边形图层。Map Layers 面板也会包括 Map 视图部分，它可以让我们将地图放大或缩小或者使之居中。

下一步就是修改地图的标题。右击标题占位符(Map Title)并选择“Title Properties”。在“Map Title Properties”对话框中，我可以修改标题为“Sales over \$1 Million in Mainland United States”。接下来，我们将修改多边形层的几个属性，以使它能显示分析数据。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

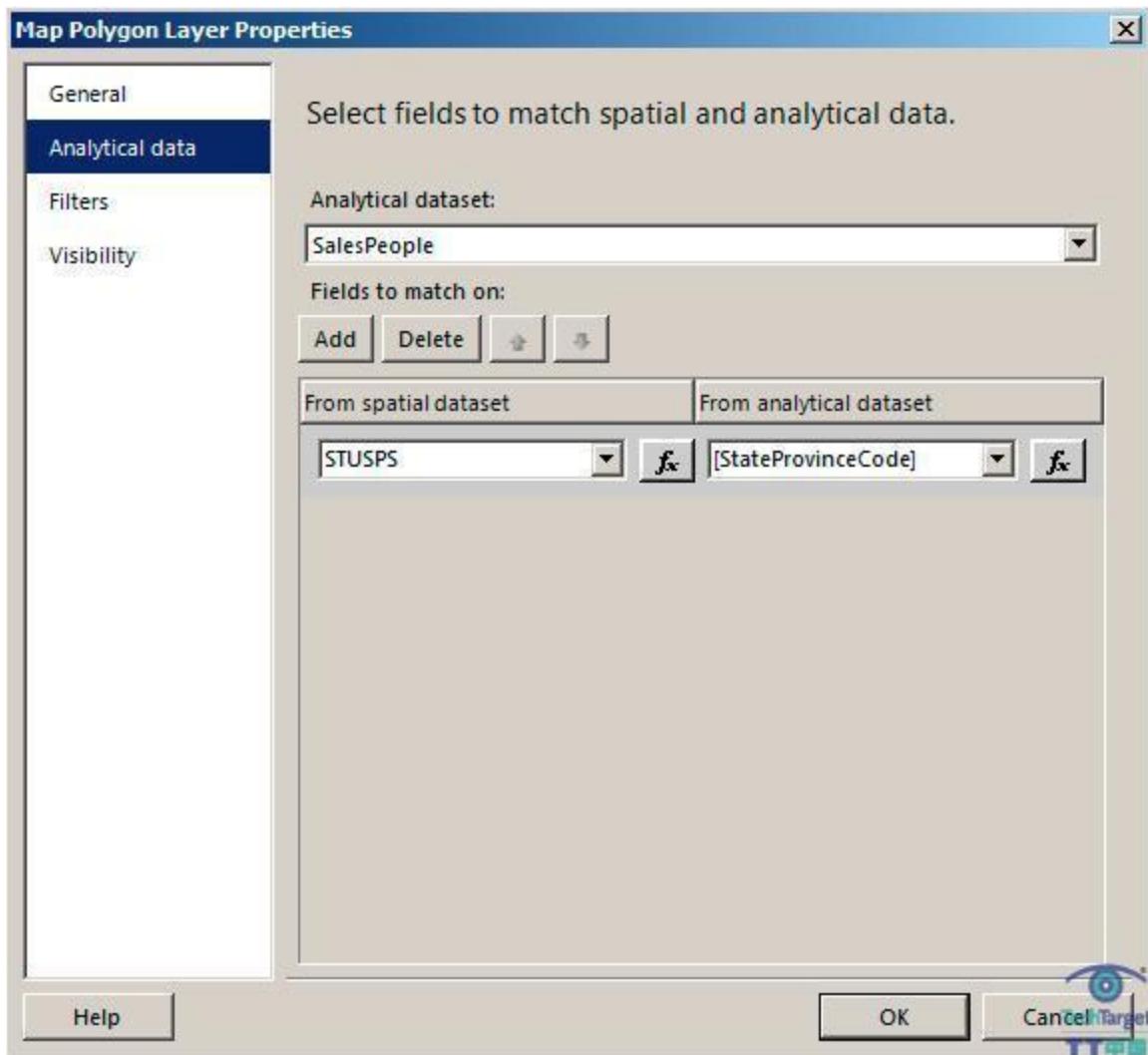
原文标题: 在 Report Builder 3.0 中添加一个多边形图层到地图

链接: http://www.searchdatabase.com.cn/showcontent_32183.htm

在 Report Builder 3.0 中修改多边形图层的属性

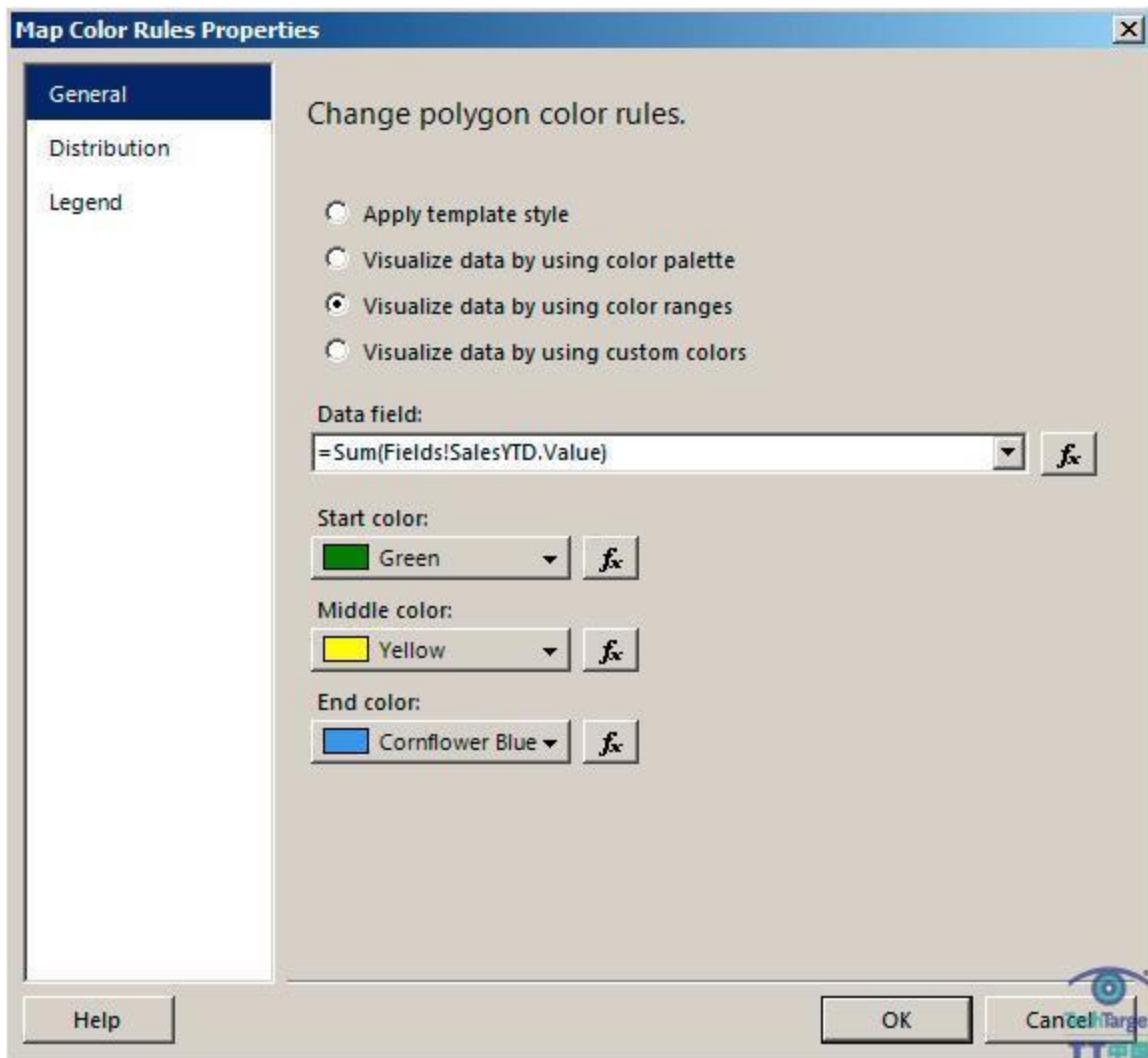
首先，让我们修改图层属性。点击“Map Layers”面板中图层的向下箭头，然后选择“Layer Data”。在“Map Polygon Layer Properties”对话框中，选择“Analytical”数据页。这个页的目的是匹配你的分析数据源到 Map Gallery 报表中的空间数据。Map Gallery 除了提供创建多边形所需要的空间数据，也包括了空间数据的详细信息，如州名。我们的目标是匹配 Map Gallery 的州代码与我们从 SQL Server 的数据集中查询得到的代码。这样，Map Gallery 的各个州可以被链接到我们的结果集中的各个州。

选择“SalesPeople”作为分析数据集，然后点击“Add”按钮就可以进行匹配。在空间数据集(Map Gallery)，选择下拉列表的“STUSPS”。其中有用于标识各个州的 U.S. Postal Service 代码。然后，在分析数据集(SalesPeople)中点击下拉列表的“StateProvinceCode”域。你应该还记得我们在数据集中所使用的查询，我去掉了 StateProvinceCode 字段的名称。如果我没有那样做，各值会多一个空格，而两个源的州代码将不能匹配。图 2 显示的配置了属性的 Analytical 数据页面。



在 Analytical 数据页面配置好后，选择 Visibility 页并将 Transparency 属性设为 40。然后点击“OK”关闭对话框。

下面我们将配置颜色规则，使州的颜色能反映该州里每一个销售人员的销售量。这些颜色将对应于基于销售客划分的不同的范围值。点击 Map Layers 面板中图层旁边的向下滑头，然后选择“Polygon Color Rule”，就可以配置颜色规则。在“Map Color Rules Properties”对话框的 General 页面，选择“Visualize data by using color ranges”选项。当选择这个选项时，会有几个属性出现，如图 3 所示。

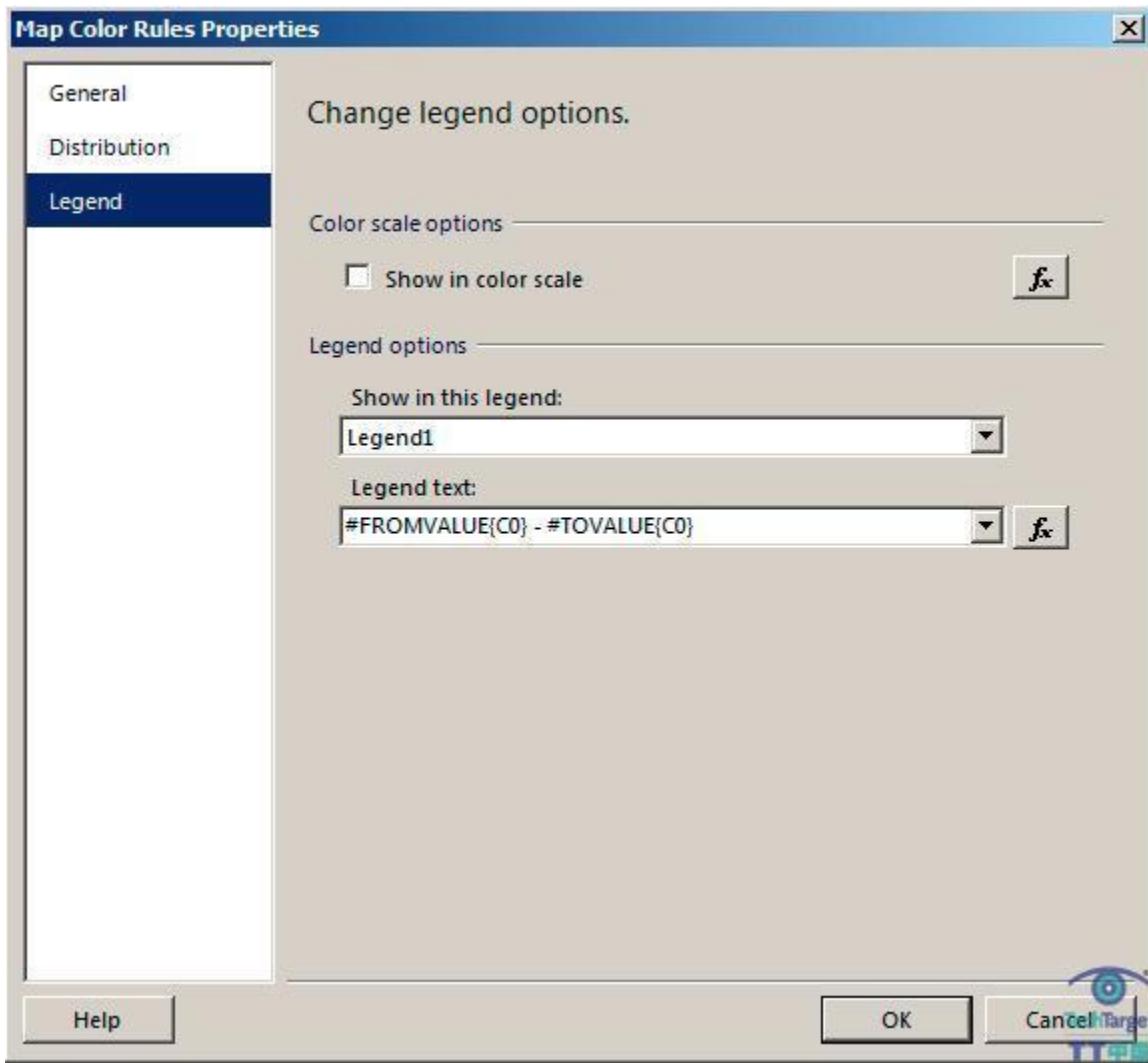


对于 Data 域属性，我选择了 [Sum(SalesYTD)]，开始颜色为绿色，中间颜色为黄色，结束颜色为浅蓝色。当然，你可以任意选择最适合你的颜色。设置颜色的最佳方法是试验不同的颜色，然后在渲染的报表中查看设置的效果。

接下来，我们将配置 Legend 页的属性，这样默认的图例能显示关于销售量和这些数量对应颜色的数据。首先，清除“Show in color scale”选择，然后选择“Legend1”作为图例。接着，我们需要修改“Legend Text”属性的表达式。这是一个.NET 表达式，它指定文本的格式。我们所需要做的就是将 N0 修改成 C0，这样数量会显示为货币值。然后这个表达式将显示为：

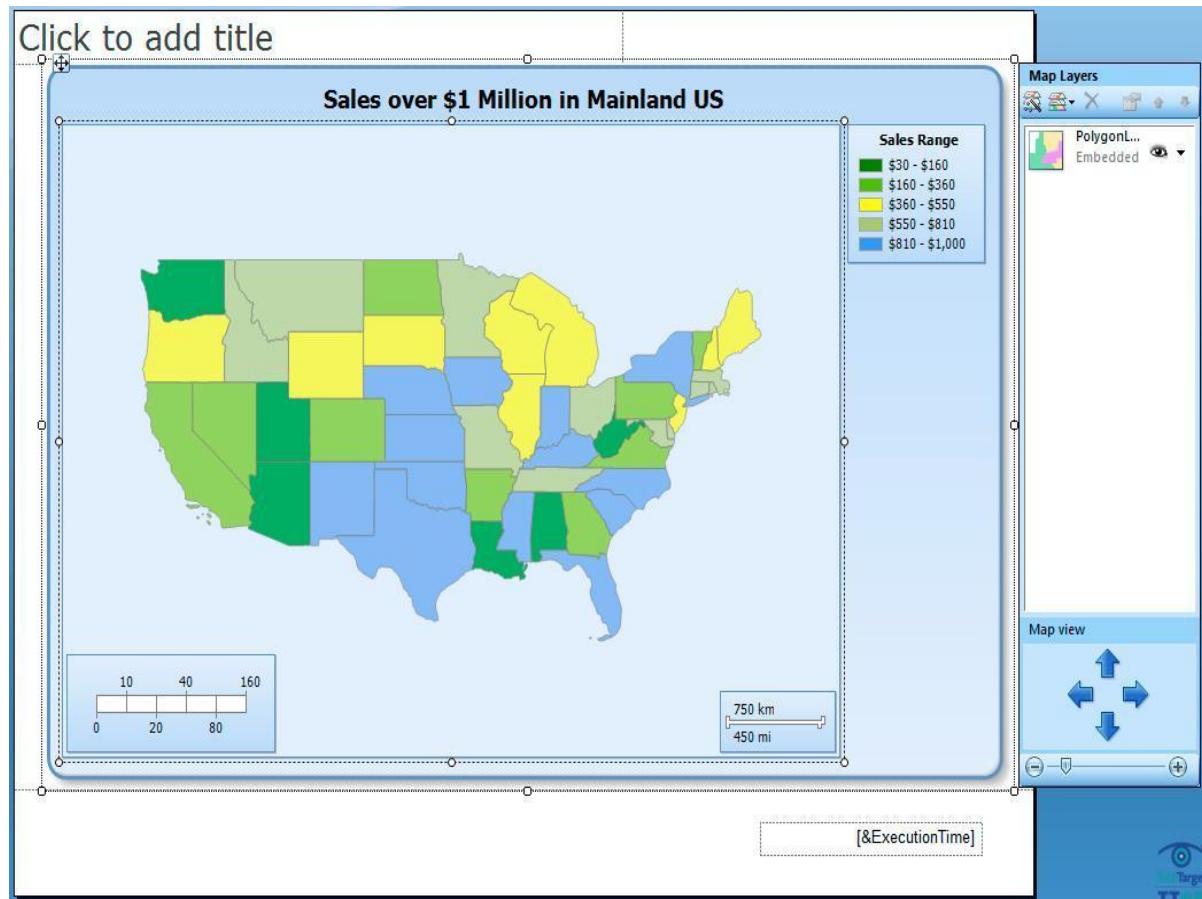
```
#FROMVALUE {C0} - #TOVALUE {C0}
```

图 4 显示了图例页面，它会显示修改后的属性值。

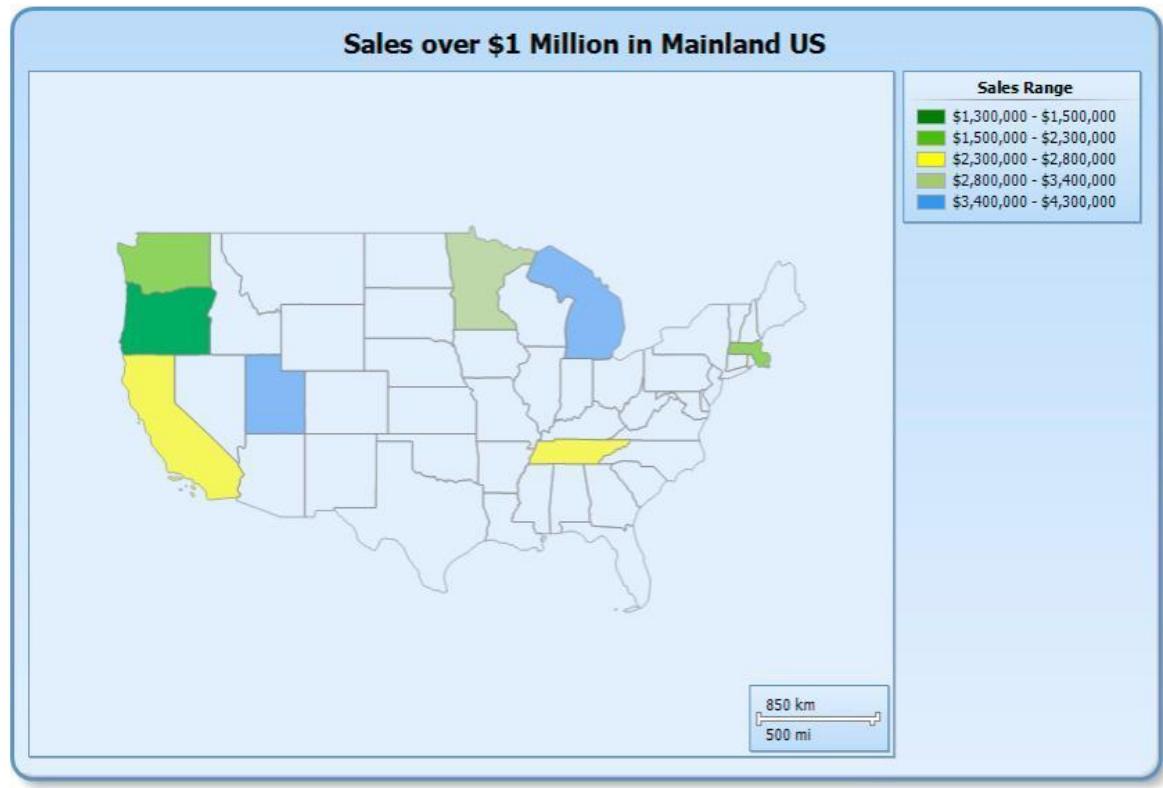


最后，我们必须配置多边形属性，这样只有关联了销售额的州才会被着色。点击“Map Layers”面板中图层旁边的向下箭头，然后选择“Polygon Properties”，就可以修改多边形的填充颜色。在“Map Polygon Properties”对话框的“Fill”页面，将颜色修改为“No Color”。

下一步就是修改图例标题。右击标题，点击“Legend Title Properties”，然后修改标题。我将我的标题修改成“Sales Range”。现在，查看设计视图中的多边形图层，它应该类似于图 5 的效果。



我们可以看到，图表显示了数量的范围和颜色刻度。各个州的颜色应该与值范围相匹配。然而，当在设计视图中查看地图时，它使用的是示例数据，所以所有的州都被着色了，即使大多数的州是没有数据的。如果要查看地图的实际外观，我们需要执行该报表。我的执行结果如图 6 所示。



2/5/2010 6:08:43 PM
 TechTarget
TT 中国

注意“Sales Range”图例现在反映了超过1百万美元销售额的实际范围，并且标尺里的颜色与位于范围内的州的颜色相匹配。我们可以看到，我们已经能够在一个图层里整合分析数据与空间数据。然而，通过添加额外的图层，我们可以在我们的地图中增加更多的细节。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 在 Report Builder 3.0 中修改多边形图层的属性

链接: http://www.searchdatabase.com.cn/showcontent_32185.htm

完成 Report Builder 3.0 中地图点的创建

当创建报表和添加多边形图层后，我们几乎已经完成了地图的创建。最后的步骤包含另外 2 个图层来完成 Report Builder 3.0 地图的创建——点和块图层。第三部分介绍这些图层所需要的设置。

我们将添加一个点图层来标记销售人员所在的位置。这个图层将基于我们从数据集中查询得到的数据。点击“Map Layers”面板顶部的“New Layer Wizard”图标，开始创建点图层。当“New Map Layer”向导启动后，选择“SQL Server spatial query”作为空间数据源。

在下个界面中，选择“SalesPeople”作为数据集。接着是提示我们选择空间域和图层类型。这个域应该是 SpatialLocation 字段，并且图层类型应该是“Point”(这两个选项默认应该已经选择了)。我们也将创建一个基本的标记地图——默认的地图可视化选择。

当我们来到提供选择颜色主题和数据可视化的提示的界面时，我们将选择“PushPin”作为标记类型，但是其它设置保留它们的默认值，最后点击“Finish”。这样点图层将被添加到地图上，并被列在“Map Layers”面板中。

下一步是配置新图层的点属性。点击“Map Layers”面板中点图层旁边的向下箭头，然后选择“Point Properties”。在“Map Point Properties”对话框的“General”页面，我们将配置 2 个属性为表达式。

在“Label”文本属性上，打开“Expression”对话框(点击表达式按钮)并填写下面的表达式：

```
=Fields!City.Value + ", " + Fields!StateProvinceCode.Value
```

该表达式连接城市名和州代码，这样它们会显示为地图上的图钉标记。

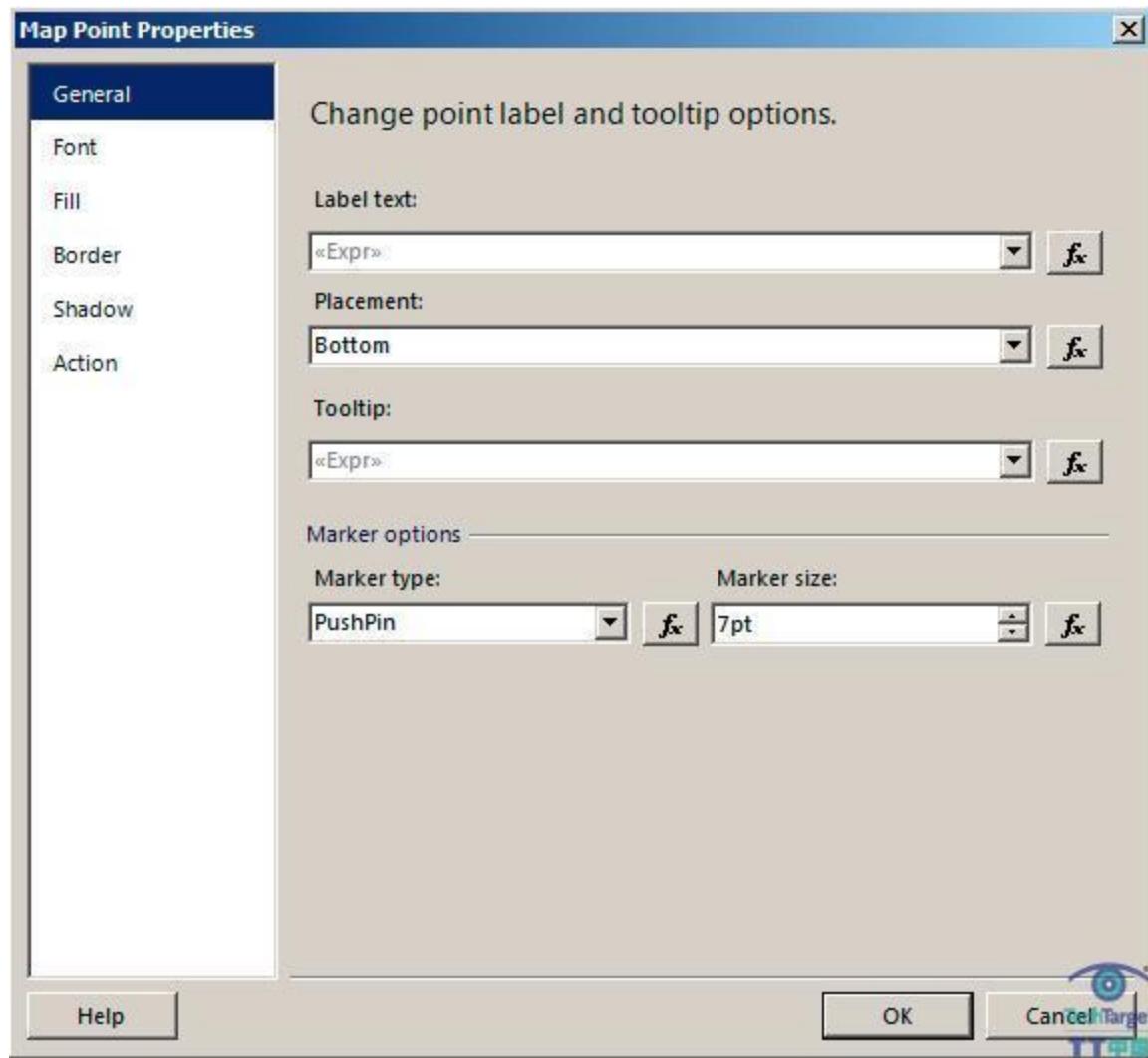
在“Tooltip”属性上，打开“Expression”对话框并填写下面的表达式：

```
=Fields!FullName.Value + ":" +
```

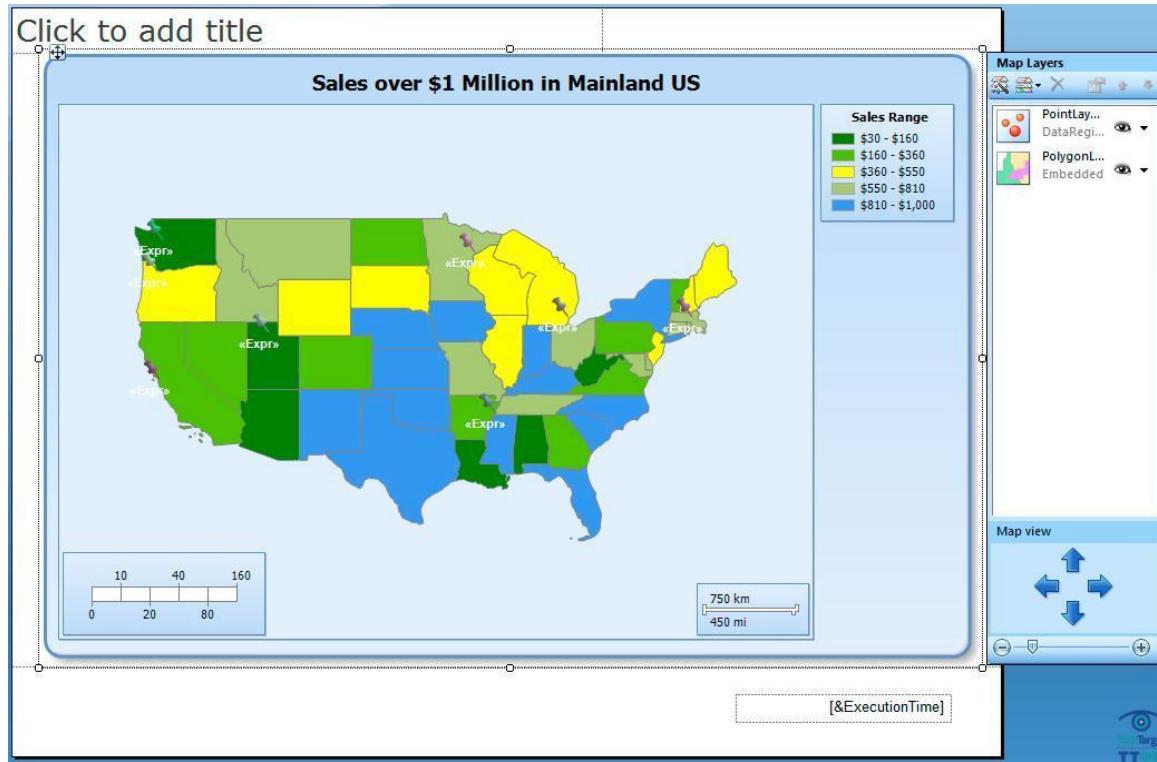
FormatCurrency(Sum(Fields!SalesYTD.Value))

这个表达式连接销售人员的姓名和销售额，这样它们会在鼠标悬停在该标记上时显示。

最后，将“Marker”大小属性减少为 7 个点距。“Map Point Properties”对话框如图 7 所示。



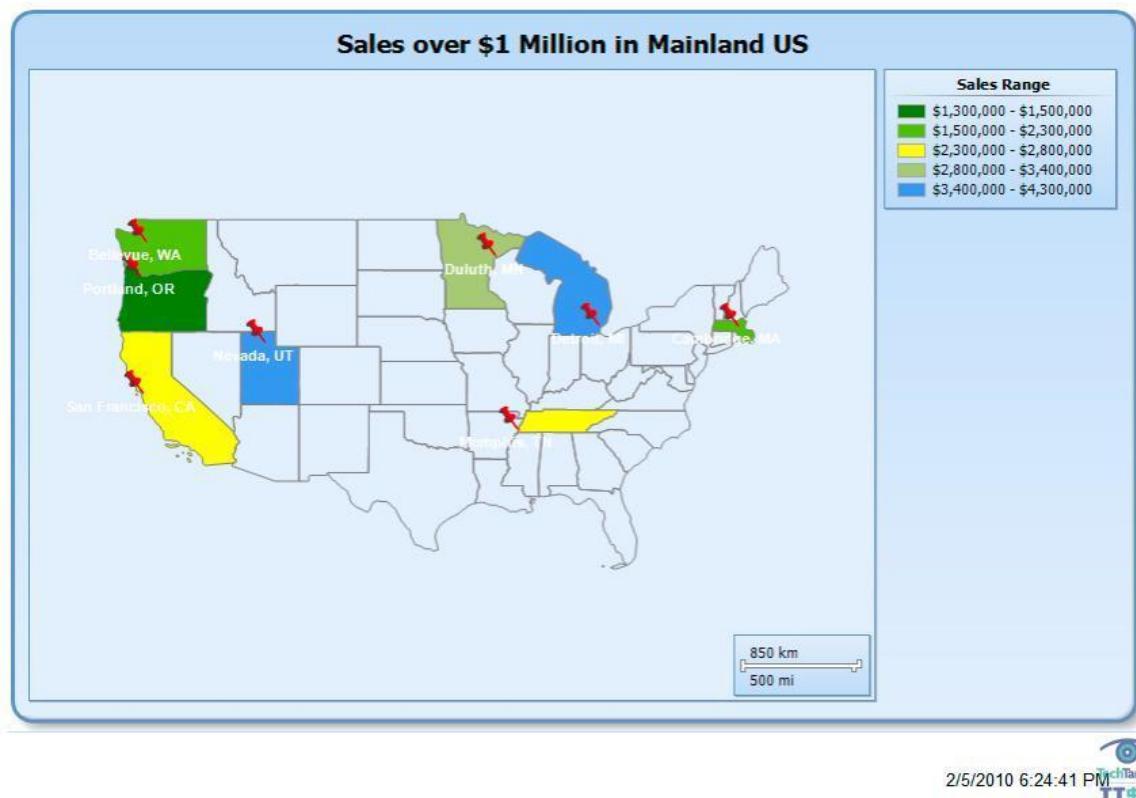
在修改了“General”页面的属性后，我们将修改“Font”页面的设置。在本例中，我使用 8 号、Arial、粗体白色字体。这就是所有需要添加到地图图层的设置了。点和多边表图层现在应该看起来如图 8 所示效果。



我们注意到现在出现了图钉标记，以及`<>`占位符，这表示这个标签是基于一个表达式的。

我们还注意到州的颜色在我们添加和配置了多边形图层后变得比之前更暗些(因为我们之前将“Transparency”属性设置为40)。然而，所添加的另一个图层这个属性被重新设为0。结果，我们需要在添加完所有图层后重新设置这个可见性。(通常，我会在我添加完所有图层后再设置这个属性，但是我只是想演示这个特定的Report Builder 3.0特性。)

图9显示报表现在报告后的样子。我们可以看到，图钉标记已经和城市和州名称一起被添加上地图上。



如果鼠标悬停在一个图钉标记上，我们可以看到销售人员的名称以及销售总额。同时，你可能注意到在当前的渲染结果中，标签是很难读的。但不要担心，我们添加的下一个图层会更暗些，那么让我们继续往下操作。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 完成 Report Builder 3.0 中地图点的创建

链接: http://www.searchdatabase.com.cn/showcontent_32241.htm

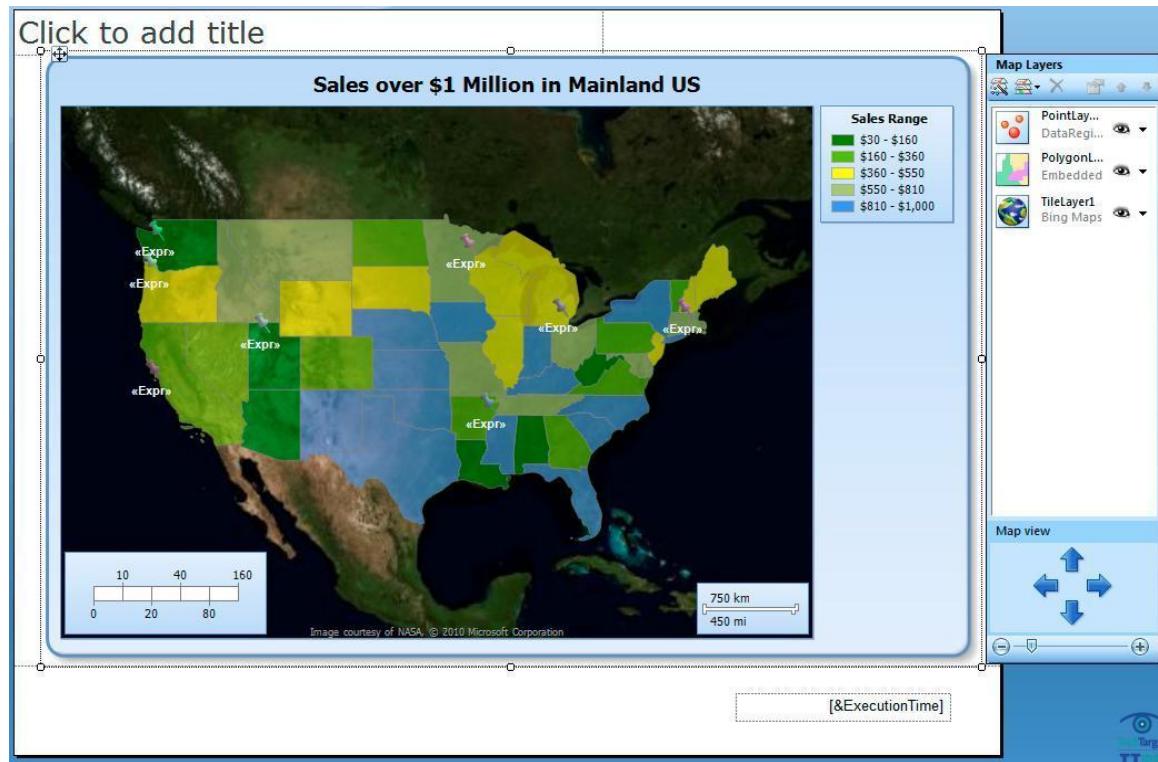
用 Report Builder 3.0 添加一个块图层到地图

添加一个块图层到地图

最后一个图层是“Virtual Earth”背景，这会令整个报表看起来更生动。我们不能使用“New Layer”向导来添加一个块图层到地图上。而是要点击“Map Layers”面板顶部的“Add Layer”图标，然后选择“Tile Layer”。这个图层会被添加到地图中并列在“Map Layers”面板中。

默认，块图层是配置为道路(Road)类型的，但是我们想要使用一个空中(Aerial)地图。点击“Map Layers”面板中块图层旁边的向下箭头并选择“Tile Properties”，修改块图层的类型。在“Map Tile Layer Properties”对话框的“General”页面上，将“Type”属性设置为“Type”，然后点击“OK”。

现在我们需要回去解决一些可见性问题。打开多边形图层的图层属性，然后在“Visibility”页面，将“Transparency”属性设为40。我们的地图在设计视图中现在会变成图10所示的效果。



我们可以看到，多边形和点图层位于块图层之上，而透明度使一些图层有透视效果。最后，我们完成的报表效果如图 11 所示。



2/5/2010 6:33:21 PM
TechTarget
TT 中国

所有 3 个图层现在被合并到一个地图中，显示了链接空间数据和分析数据的无缝数据源，而图例则提供销售范围的快速预览。当然，这只是个基本的地图。我们还可以使用其它特性来使地图变得更加丰富，这包括连接点的线条图层。尽管如此，我们应该已经了解了 Report Builder 3.0 的地图功能大大地丰富其数据可视化工具集，这是一个可以使各种报表更加生动和美观的方法。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: 用 Report Builder 3.0 添加一个块图层到地图

链接: http://www.searchdatabase.com.cn/showcontent_32243.htm

Report Builder 3.0 矩阵初探

在 Report Builder 3.0 中创建一个报表时，我们可能想要基于分组数据显示汇总信息。例如，假设我们计划创建一个报表，显示各产品线的合计销售总额、产品和销售地点。为了显示诸如这里列举的汇总数据，Report Builder 3.0 使用的是 矩阵。

矩阵 是一种表，它按行、列或行与列显示汇总数据 —— 类似于交叉表或 数据透视表。在列组或行组中的唯一行的数量决定了矩阵报表返回的行和列的数量。一个矩阵可以包括多个行和列组，而这些组可以按层次排列，以显示有意义的数据分类和子分类。

那么让我们看一下如何添加一个矩阵到一个 Report Builder 3.0 报表中。在这个例子中，矩阵将从 AdventureWorks 示例数据仓库查询销售数据。列组将基于销售组和国家生成。行组则基于订单生成的年份、季度和月份生成。汇总数据将基于实际的销售额生成。

创建数据源和数据集

在添加一个矩阵到一个报表之前，我们应该先创建一个数据源和数据集。对于本文所演示的矩阵，我创建了一个连接到本地 SQL Server 2008 R2 的一个实例上的 AdventureWorks2008R2 示例数据库的数据源。我将数据源命名为数据库名。

注意：我在 Windows Server 2008 上安装了 SQL Server 2008 R2 的 2009 年 11 月社区技术预览版(CTP)的一个本地实例。这个安装包括数据引擎和 SQL Server Reporting Services (SSRS)。我还安装了 SQL Server 2008 R2 版本的 AdventureWorks 示例数据库。Report Builder 3.0 的 11 月 CTP 版本也安装在相同的服务器上。

接下来，我创建了一个名为 InternetSales 的数据集。这个数据集使用下面的 Transact-SQL 语句通过 AdventureWorks2008R2 数据源查询数据：

```
SELECT
    t.SalesTerritoryCountry SalesCountry,
    t.SalesTerritoryGroup SalesGroup,
    d.MonthNumberOfYear MonthNumber,
    d.EnglishMonthName OrderMonth,
```

```
d. CalendarQuarter OrderQuarter,  
d. CalendarYear OrderYear,  
s. SalesAmount  
FROM  
FactInternetSales s  
INNER JOIN DimSalesTerritory t  
ON s.SalesTerritoryKey = t.SalesTerritoryKey  
INNER JOIN DimDate d  
ON s.OrderDateKey = d.DateKey
```

这个语句查询销售区域、销售团队和订单发生日期等销售数据。在我们创建好数据集后，就可以添加矩阵到报表中了。然而，注意矩阵通常是报表中其它元素内的一个元素，但本文我只关注于添加矩阵。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

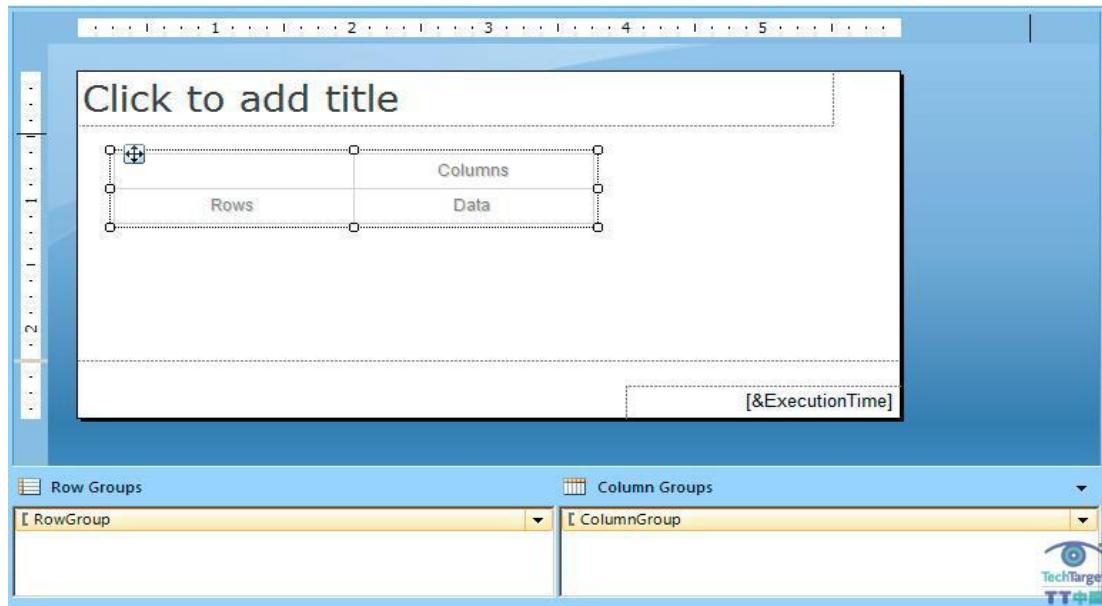
原文标题: Report Builder 3.0 矩阵初探

链接: http://www.searchdatabase.com.cn/showcontent_33158.htm

Report Builder 3.0 创建矩阵报表：准备工作

一旦创建了数据源和数据集创建，我们就可以添加新的矩阵表到报表中。本文演示在 Report Builder 3.0 中如何开始创建矩阵，并详细介绍添加到表中的层次和总数额。

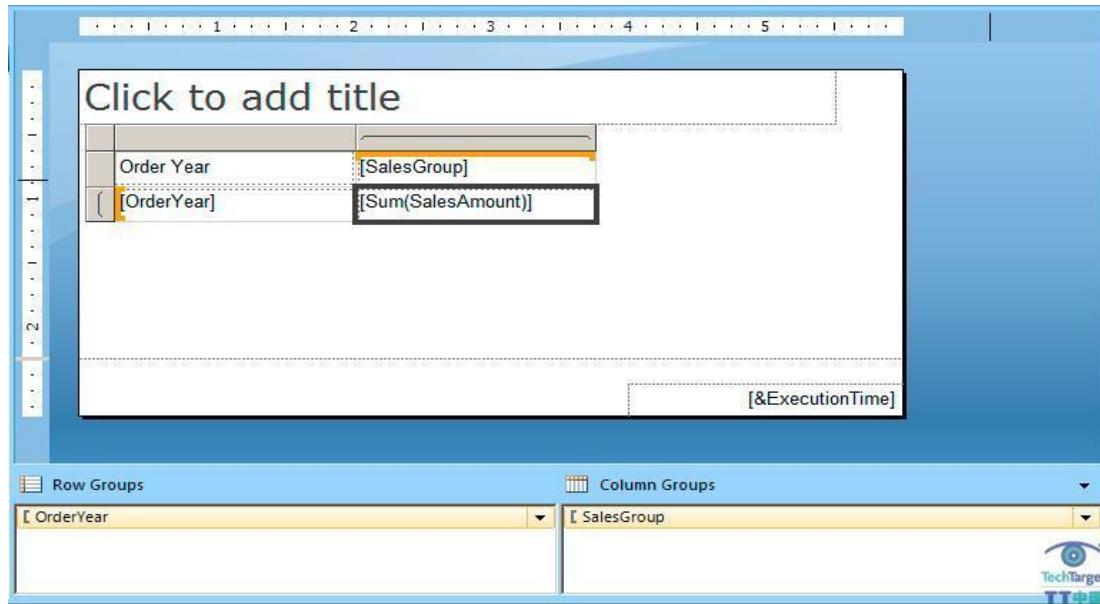
在我们定义了数据源和数据集后，我们就可以添加矩阵到报表中。在“Insert”菜单，点击“Matrix”按钮的向下箭头，然后选择“Insert Matrix”。接着，将鼠标指针移到报表设计界面。鼠标指针会变成一个旁边有小图标表的“+”号。这表示这是矩阵的左上角。点击设计界面然后往右下拖曳鼠标画出表的形状。释放鼠标按钮，设计界面的显示应该类似于图 1。



这个时候不要担心行或列字段的大小设置。在行和列组添加后，设置表的大小和配置矩阵属性会更加容易。

第一步是从 InternetSales 数据集中将 SalesGroup 字段拖到矩阵的“Columns”位置。数据集显示在左边面板中，在数据集节点下面。如果 InternetSales 数据集和它的字段没有显示，可以在需要时展开“Datasets”节点。SalesGroup 字段就成为矩阵中的第一个列字段组。

接下来，将 OrderYear 字段拖到矩阵的“Rows”位置。这样 OrderYear 就成为矩阵的第一个行组。最后，将 SalesAmount 字段拖到“Data”位置。SalesAmount 字段包含了报表渲染时汇总的数据。当添加字段时，会自动创建一个汇总表达式。现在我们的矩阵应该类似于图 2 所示。



在我们添加元素到矩阵后，通过预览数据来确保数据符合你的要求是很好的做法。点击“Home”菜单上的“Run”按钮，就可以预览矩阵。然后设计界面将切换到查看界面，矩阵将会显示，如图 3 所示。

Order Year	Europe	North America	Pacific
2005	709947.2016	1247379.2572	1309047.1978
2006	1627759.7146	2748298.9283	2154284.8835
2007	3382979.2672	3374296.8174	3033784.2131
2008	3209356.0800	3997659.3700	2563884.2900

我们可以看到，数据是根据年份(行组)和地域(列组)进行分组的。对于每对年份和地区，都显示一个汇总的销售总额。例如，2005 年 Europe 的销售总额为 709947.2016。(如之前所提到的，我们将在添加完所有行和列组再配置矩阵的属性。)

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: Report Builder 3.0 创建矩阵报表: 准备工作

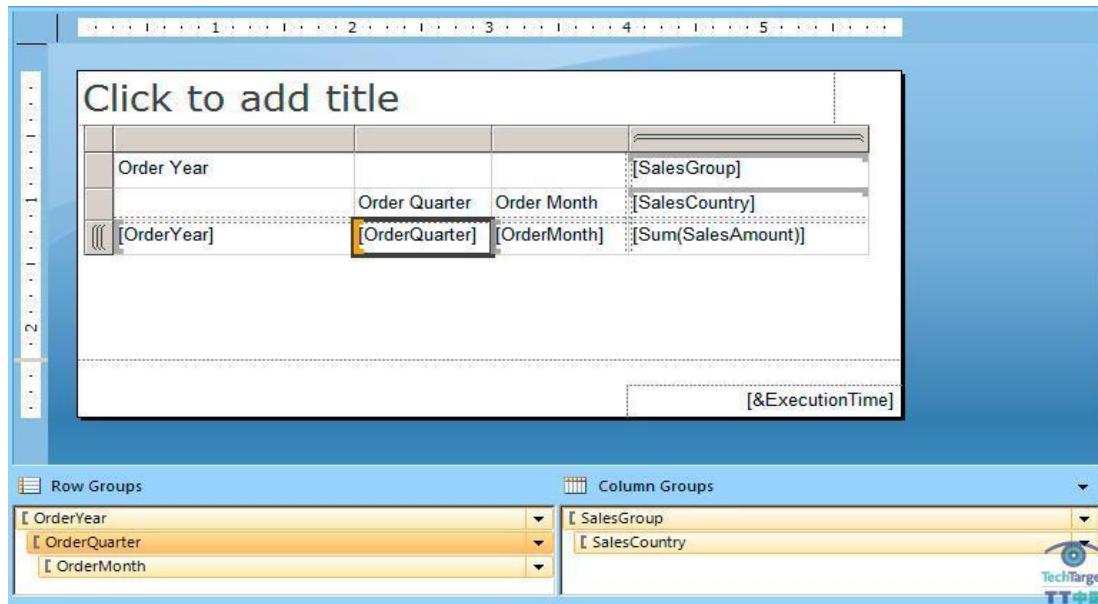
链接: http://www.searchdatabase.com.cn/showcontent_33213.htm

Report Builder 3.0 创建矩阵报表：添加组层次

虽然目前创建的矩阵显示了有用的信息，我们可以通过添加额外的组来完善报表。其中添加组最快速的方法是创建 层次。例如，现在已经有的地区，都是对应于销售团队的。然而，销售团队可以按照国家划分。这样，我们就能得到一个自然的层次，其中 SalesGroup 字段作为 SalesCountry 字段的上级。如果矩阵反映了这样的层次，汇总数据将会更加有层次，从而更加有用。

从 InternetSales 拖曳 SalesCountry 属性到 [SalesGroup] 单元格，就可以创建这个层次。通过拖曳字段创建层次的技巧是保证层次应该流向的方向对应于单元格加粗边框所表示的(这在实际使用时会更有意思)。换句话说，当我们将 SalesCountry 字段拖到矩阵的 [SalesGroup] 单元格时，[SalesGroup] 单元格的底边框会变成加粗的。完成后，释放鼠标按钮。然后新的 [SalesCountry] 单元格会插入到 [SalesGroup] 单元格旁边。

创建日期层次的方法也是一样的。在本例中，我们想要一个在顶部包含 OrderYear 的层次。从 InternetSales 数据集拖曳 OrderQuarter 字段到矩阵的 [OrderYear] 单元格，就可以创建这个层次。在放开 OrderQuarter 字段之前，要保证 [OrderYear] 单元的右边框是加粗的。然后再对 OrderMonth 字段进行相同的操作。将它拖到 [OrderQuarter] 单元格，然后在右边框加粗时放开它。我们的矩阵现在应该类似于图 4 所示。



您可能已经注意到现在行和列组已经列在底部的两个面板上，其中还有我们所定义的层次。我们将在后面看到，我们可以修改组属性来改变它们的行为以及它们的显示方式。

在我们添加了组和创建层次后，我们应该再一次预览数据。图 5 显示了添加了两个新的层次后的矩阵。（当报表渲染后，整个矩阵现在会包含更多的列和行。）

Order Year	Order Quarter	Order Month	Europe			North America	
			France	Germany	United Kingdom	Canada	United States
2005	3	August	18208.8882	40759.1664	54334.1264	28918.6982	141432.5228
		July	40174.3264	35604.4200	37498.4346	21965.4382	128492.6392
		September	36278.5182	22168.7182	62888.8628	17891.3500	160722.0692
	4	December	37676.7146	55682.3228	54487.4064	36278.5182	297217.3838
		November	36799.3364	43396.0364	26966.9046	29846.0764	196301.4892
		October	11433.9082	40174.3264	55414.7846	11929.7264	176383.3456
	2006	1 February	30316.8946	47712.4264	47177.5864	161136.4082	100312.2974
		January	33246.0664	9075.5546	71298.9982	50413.3182	209604.4120
		March	42068.3410	61123.1282	36100.2382	35579.4200	236679.0956
		2 April	40759.1664	53869.7874	35376.1400	67783.8500	241452.2820
		June	35496.6410	55033.2246	42900.2182	65222.2164	268451.9656
		May	51786.5146	48754.0628	26064.5264	46135.9500	295126.5684
		3 August	41607.2059	34224.0046	64033.4327	67618.4817	188777.8866
		July	42458.3978	39267.0010	42599.0152	51652.5588	193398.4360
		September	48291.0266	38210.1574	51904.8969	17417.7571	78392.3613
	4 December	77967.6876	52372.3626	76087.2282	23858.0856	141075.5222	
		November	26180.7671	50322.8479	47944.8149	9231.0314	97976.1943
		October	44763.3041	31266.2900	50100.6587	25553.3049	75449.5246
2007	1 February	41060.2120	64197.1196	63469.1880	17352.8827	95228.8969	
		January	40726.2555	37915.4438	61743.9801	16287.4142	95953.8504
		March	65006.9463	52867.2501	61714.4480	28248.5765	78949.0861
	2 April	59362.0606	73408.1122	49171.9244	11943.9353	107160.1578	
		June	91487.8290	63781.3150	57193.6990	5686.4978	134014.6844
	May	54488.0058	55415.3398	70631.6880	30160.1659	147919.7197	TT中国

我们可以看到，行组按照年份、季度和月份组织到一个层次中。每年被分成恰当的季度(只包含有销售的季度)，而每个季度会分成几个月份。而列组也被除数组织成地区层次，同时每一个销售团队按国家划分。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: Report Builder 3.0 创建矩阵报表: 添加组层次

链接: http://www.searchdatabase.com.cn/showcontent_33215.htm

Report Builder 3.0 创建矩阵报表：添加总数到矩阵

注意在图 5 中，每一个层次的最低级上有一个汇总的销售总额。而且，对层次的其它级别进行总数统计也是很不错的。例如，我们可能想要看到每年的总数和所有年份的总数。

Report Builder 3.0 有一个功能，它很容易给矩阵添加汇总列。例如，右击矩阵中的[OrderYear]单元格，选择“Add Total”，然后点击“After”，就可以添加一个总数到所有行的末尾。添加到矩阵的行在第一个单元格显示单词“Total”，而在地区层次旁边的单元格同显示合计表达式。现在对于[OrderQuarter]单元格重复这个相同的过程。这次[OrderQuarter]单元格这边会插入一行，但是它是在[OrderYear]单元格的右边。最后，在[SalesGroup]单元格中重复这个过程。注意，因为这是一个列组，被添加的是新列，而不是行，如图 6 所示。

现在如果预览报表，我们将看到行总数已经添加到每一个季度(年份的末尾)以及所有年份后面。此外，所有数据的右边增加了一个列。这一列显示每一行的汇总销售总数。图 7 显示了矩阵的一部分，以及每一组季度的总数。

Order Year	Order Quarter	Order Month	Europe			North America	
			France	Germany	United Kingdom	Canada	United States
2005	3 August	18208.8882	40759.1664	54334.1264	28918.6982	141432.5228	
		July	40174.3264	35604.4200	37498.4346	21965.4382	128492.6392
		September	36278.5182	22168.7182	62888.8628	17891.3500	160722.0692
	4 December	37676.7146	55682.3228	54487.4064	36278.5182	297217.3838	
		November	36799.3364	43396.0364	26966.9046	29846.0764	196301.4892
		October	11433.9082	40174.3264	55414.7846	11929.7264	176383.3456
	Total		180571.6920	237784.9902	291590.5194	146829.8074	1100549.4498
2006	1 February	30316.8946	47712.4264	47177.5864	161136.4082	100312.2974	
		January	33246.0664	9075.5546	71298.0982	50413.3182	209604.4120
		March	42068.3410	61123.1282	36100.2382	35579.4200	236679.0956
	2 April	40759.1664	53869.7874	35376.1400	67783.8500	241452.2820	
		June	35496.6410	55033.2246	42900.2182	65222.2164	268451.9656
		May	51786.5146	48754.0628	26064.5264	46135.9500	295126.5684
	3 August	41607.2059	34224.0046	64033.4327	67618.4817	188777.8866	
		July	42458.3978	39267.0010	42599.0152	51652.5588	193398.4360
		September	48291.0266	38210.1574	51904.8969	17417.7571	78392.3613
	4 December	77967.6876	52372.3626	76087.2282	23858.0856	141075.5222	
		November	26180.7671	50322.8479	47944.8149	9231.0314	97976.1943
		October	44763.3041	31266.2900	50100.6587	25553.3049	75449.2446
	Total		514942.0131	521230.8475	591586.8540	621602.3823	2126696.5460

TT中国

我们可以看到，合计的总数已经自动生成。可惜，这些总数并不总是很容易辨别。事实上，区别不同季度的总计是很困难的，特别是当我们查看报表的右边。因此，我们应该配置矩阵，保证数据以一种更容易理解的方式显示。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: Report Builder 3.0 创建矩阵报表: 添加总数到矩阵

链接: http://www.searchdatabase.com.cn/showcontent_33217.htm

Report Builder 3.0 中矩阵配置技术

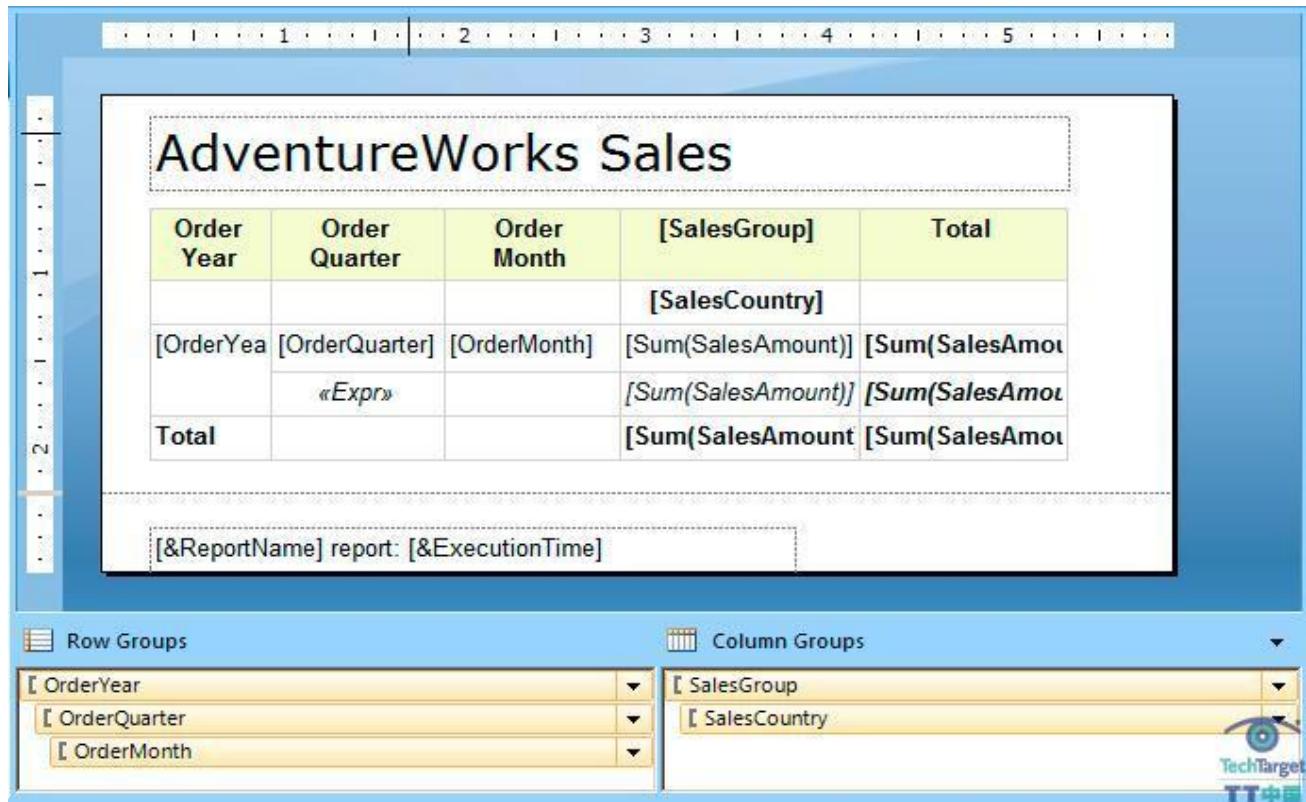
虽然我们已经添加了一个矩阵表到报表中 —— 但这还不够。本部分阐述了一些使用 Report Builder 3.0 配置矩阵的技巧，以保证数据是以预期的方式显示。

现在我们已经添加组到矩阵了，我们可以调整单元格的大小和配置属性了。我们可以在任何时间进行修改，但是我发现在矩阵填充了所有数据后进行配置效率会更高。当然，我们可以在需要时添加和删除组，但是至少我们已经能够开始创建了。

选择矩阵，这样上边和左边的屏幕的边框变成加粗灰色(如图 6 所示)，现在我们就可以修改单元格的大小了。移动鼠标到边框的适当位置，然后根据需要修改单元格大小。我们通过修改单元格来练习修改单元格大小，然后再预览这个报表。这样我们就能够确定以我们想要的方式显示数据。

现在让我们看一下如何重新配置属性。对于本例，我们一次只修改一行。第一行显示列标题。根据我们添加列和行组的顺序，标题可能不会总是显示在第一行。然而，我们的这个报表的第一行就是标题。列标题应该对应于数据。所以第一列是“Order Year”，第二列是“Order Quarter”，而第三列是“Order Month”。第四列实际上是一个表达式 —— [SalesGroup]。这表示销售团队的名称将会作为标题插入。第五列应该是“Total”。

接下来，选择第一行的所有单元格，然后点击“Home”菜单的“Center”和“Bold”按钮。现在，保持单元格的选择，在屏幕右边的“Properties”面板设置“BackgroundColor”属性。我们可以选择喜欢的颜色。我这里选择的是黄绿色，代码是：f7fccf。我们的第一行现在应该类似于图 8 所显示的样子。(注意图 8 显示的是我们完成所有格式修改后矩阵的外观。您也可以使用这个图来指导完成本文后面所有的配置。)



The screenshot shows a Microsoft Report Builder interface. At the top, there is a navigation bar with page numbers 1 through 5. Below the navigation bar is a title box containing the text "AdventureWorks Sales". The main content area displays a matrix report. The matrix has five columns: "Order Year", "Order Quarter", "Order Month", "[SalesGroup]", and "Total". The first row of the matrix contains the labels "[OrderYear]", "[OrderQuarter]", "[OrderMonth]", "[Sum(SalesAmount)]", and "[Sum(SalesAmount)]". The second row contains the label "[SalesCountry]". The third row contains the label "«Expr»". The bottom row contains the label "Total". Below the matrix, there is a text box containing the text "[&ReportName] report: [&ExecutionTime]". At the bottom of the interface, there are two panes: "Row Groups" and "Column Groups". The "Row Groups" pane shows "OrderYear", "OrderQuarter", and "OrderMonth". The "Column Groups" pane shows "SalesGroup" and "SalesCountry".

矩阵中的第二行应该在其中一个单元格中显示标签——[SalesCountry]单元格。和第一行一样，文本的格式是居中和粗体。如果其它单元格还有标签，我们必须删除它们。

在第三行上，选择除[OrderYear]之外的所有单元格。接着，在“Properties”面板上点击与“BackgroundColor”属性相关联的向下箭头，然后选择“Expression”。在“Expression”对话框中输入下面的表达式：

```
=Iif(Fields!OrderQuarter.Value = 1 Or Fields!OrderQuarter.Value = 3, "Cornstalk", "Ivory")
```

这个表达式使用“Iif”函数来设置行的颜色。如果“OrderQuarter”字段的值是1或3，它的颜色会被设为“Cornstalk”。否则，颜色是“Ivory”。结果，偶数季度的行颜色将与奇数的不同。我所做的另一个修改是居中显示[OrderQuarter]单元格的文本。

在第四行，选择除[OrderYear]之外的所有单元格，然后将类型设置为斜体。然后右击[OrderQuarter]单元格，选择“Expression”。在“Expression”对话框中，输入下面的表达式：

```
=Fields!OrderYear.Value & " Total"
```

这个表达式修改文本框的“Value”属性，这样年份会包含单词“Total”。在添加表达式后，居中显示文本。

对于第五行，直接选择所有单元格并将文本设为粗体。同时也将“Total”字段的所有单元格设置为粗体。

现在我们应该配置数据值显示为货币形式。对于每一个包含美元数额的单元格，右击单元格，然后选择“Text Box Properties”。在“Number”页，选择“Currency”作为分类，后面是“Use 1000 separator”选项。

如果不希望单独配置数字单元格，我们可以将它们作为一个组进行配置(我们需要知道正确的格式)。首先，选择所有数字单元格。在“Properties”面板中，点击与“Format”属性关联的向下箭头，然后选择“Expression”。在“Expression”对话框中，输入以下内容：

```
'$'#,0.00; ('$'#,0.00)
```

这个表达式指定这个值将显示为货币值，保留两位小数，使用逗号分隔数字。

我们应该做的另一个修改是设计月份名称的排序方式。在设计界面底部的“Row Groups”面板，点击“OrderMonth”组旁边的向下箭头，选择“Group Properties”，然后选择“Sorting”页面。在“Sort by”下拉列表中，选择“MonthNumber”字段。这样，月份将根据数字，而不是名称进行排序。

我所做的最后一个格式化修改是添加一个标题“AdventureWorks Sales”，并修改页脚，所以我会使用下面的表达式：

```
[&ReportName] report: [&ExecutionTime]
```

这样，每次报表渲染时，报表名称旁边同时会显示执行时间。现在，当我们预览报表时，它应该类似与图 9 所示效果。

AdventureWorks Sales

Order Year	Order Quarter	Order Month	Europe			North America		Pacific	Total
			France	Germany	United Kingdom	Canada	United States		
2005	3	July	\$40,174.33	\$35,604.42	\$37,498.43	\$21,965.44	\$128,492.64	\$209,652.90	\$473,388.16
		August	\$18,208.89	\$40,759.17	\$54,334.13	\$28,918.70	\$141,432.52	\$222,538.29	\$506,191.69
		September	\$36,278.52	\$22,168.72	\$62,888.86	\$17,891.35	\$160,722.07	\$173,993.51	\$473,943.03
	4	October	\$11,433.91	\$40,174.33	\$55,414.78	\$11,929.73	\$176,383.35	\$217,993.38	\$513,329.47
		November	\$36,799.34	\$43,396.04	\$26,966.90	\$29,846.08	\$196,301.49	\$210,683.56	\$543,993.41
		December	\$37,676.71	\$55,682.32	\$54,487.41	\$36,278.52	\$297,217.38	\$274,185.55	\$755,527.89
	2005 Total		\$180,571.69	\$237,784.99	\$291,590.52	\$146,829.81	\$1,100,549.45	\$1,309,047.20	\$3,266,373.66
2006	1	January	\$33,246.07	\$9,075.55	\$71,298.10	\$50,413.32	\$209,604.41	\$223,109.11	\$596,746.56
		February	\$30,316.89	\$47,712.43	\$47,177.59	\$161,136.41	\$100,312.30	\$164,161.08	\$550,816.69
		March	\$42,068.34	\$61,123.13	\$36,100.24	\$35,579.42	\$236,679.10	\$232,584.98	\$644,135.20
	2	April	\$40,759.17	\$53,869.79	\$35,376.14	\$67,783.85	\$241,452.28	\$224,451.06	\$663,692.29
		May	\$51,786.51	\$48,754.06	\$26,064.53	\$46,135.95	\$295,126.57	\$205,688.58	\$673,556.20
		June	\$35,496.64	\$55,033.22	\$42,900.22	\$65,222.22	\$268,451.97	\$209,659.38	\$676,763.65
	3	July	\$42,458.40	\$39,267.00	\$42,599.02	\$51,652.56	\$193,398.44	\$130,989.75	\$500,365.16
		August	\$41,607.21	\$34,224.00	\$64,033.43	\$67,618.48	\$188,777.89	\$149,740.46	\$546,001.47
		September	\$48,291.03	\$38,210.16	\$51,904.90	\$17,417.76	\$78,392.36	\$116,250.79	\$350,466.99
	4	October	\$44,763.30	\$31,266.29	\$50,100.66	\$25,553.30	\$75,449.52	\$188,257.15	\$415,390.23
		November	\$26,180.77	\$50,322.85	\$47,944.81	\$9,231.03	\$97,976.19	\$103,439.43	\$335,095.09
		December	\$77,967.69	\$52,372.36	\$76,087.23	\$23,858.09	\$141,075.52	\$205,953.11	\$577,314.00
	2006 Total		\$514,942.01	\$521,230.85	\$591,586.85	\$621,602.38	\$2,126,690.55	\$2,154,284.88	\$6,530,343.53
2007	1	January	\$40,726.26	\$37,915.44	\$61,743.98	\$16,287.41	\$95,953.85	\$186,238.23	\$438,865.17
		February	\$41,060.21	\$64,197.12	\$63,469.19	\$17,352.88	\$95,228.90	\$207,782.04	\$489,090.14
		March	\$65,006.95	\$52,067.25	\$61,714.45	\$28,248.58	\$78,949.09	\$190,788.49	\$485,574.79

这个图只显示部分结果。我们可以看到，现在各行都根据季度显示不同颜色，汇总销售额显示为货币格式，年度总量是斜体的，而“Total”列是加粗的。当然，我们可以进行其它的格式修改，如增加或减少组。

矩阵是一个以一种容易定制和高效渲染的方式显示汇总数据的强大工具。虽然本文只是作了基本的介绍，但是如果我们将矩阵次数越多，它会体现更多的价值。

(作者: Robert Sheldon 译者: 曾少宁 来源: TT 中国)

原文标题: Report Builder 3.0 中矩阵配置技术

链接: http://www.searchdatabase.com.cn/showcontent_33653.htm