**数据的分组与排序**

本节中将为你展示如何使用数据分组与排序，动态求和以及创建表达式等方式对报表中的数据进行有效的管理和组织。

#### 数据分组

完成基本布局的设计之后，你可以按照某些字段或者其他条件对报表内容进行分组，从而使报表阅读起来更加容易。通过分组，你可以将报表更加直观的分割成几部分从而对每个分组单独进行介绍和数据汇总。报表主要依靠分组表达式来进行分组。分组表达式主要基于一个或者多个字段集，当然你也可以根据需要设置更加复杂的分组条件。

你可以使用**C1ReportDesigner**程序或者使用代码来实现报表的数据分组：

**使用C1ReportDesigner进行分组和排序**

即使不打算显示分组的页眉和页脚部分，你也可以使用分组来对数据进行排序。你可以使用**C1ReportDesigner**对你的报表进行分组，具体位置如下图所示。

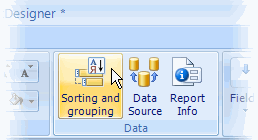


图 1

要在报表中添加或编辑分组,你需要完成以下步骤:

1. 打开**C1ReportDesigner**应用程序。更多细节，可以参阅葡萄城系列文档中的Accessing C1ReportDesigner from Visual Studio章节。（《Reports for .NET DesignerEdition》238页）

　　2. 在**Data**分组中的**Design**选项卡上单击**Sorting and Grouping**按钮。

　　单击之后，将会弹出**Sorting and Grouping** 对话框。你可以在此页面中创建、编辑、排序和删除分组。

　　3. 单击**Add**按钮，创建一个分组,并设置新分组的属性。

**Group By**字段定义数据将如何在报表中分组。对于简单的分组,你可以直接从下拉列表中选择字段。对于更复杂的分组,你可以输入分组表达式。例如,您可以使用国家字段来分组或者使用**Left(Country, 1)**表达式通过国家首字母来分组。

　　4. 本示例中，选择**Country**作为**Group By**的表达式。

　　5. 接下来，选择你想要的排序类型(本例中选择升序（**Ascending**）)。你还可以指定新分组的页眉和页脚部分是否可见，以及分组是否在同一页面上进行呈现。

**注意：** 你不能将备注型字段或者二进制(对象）字段用于分组和排序，这是OLEDB对此进行的限制。

**Sorting and Grouping**对话框效果如下图所示：

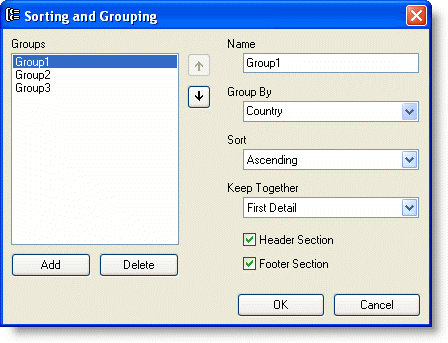


图 2

如果你添加了很多字段，可以通过**Group**列表右侧的箭头按钮改变字段的顺序。该操作将自动调整报表分组中页眉和页脚的位置。如果想要删除某个字段，可以使用**Delete**按钮。

完成字段设置之后，单击**OK**按钮关闭对话框，你可以在**Designer**中看到设置后的效果。报表中增加了页眉和页脚两个新的区域。新增区域（页眉和页脚）的高度均默认为0，你可以通过鼠标拖动边缘来扩展该区域。需要注意的是，页眉区域是可见的。而页脚区域是不可见的。因为在之前的对话框中，**Group Header**按钮已经被选中，而**Group Footer**按钮未被选中。不可见区域通过一个阴影图案表明该区域在报表中是不可见的。具体实现效果如下图所示：

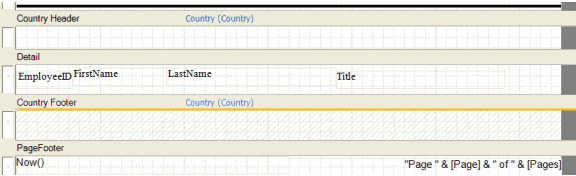


图 3

在新区域顶部的标题栏上的标签中，包含了该区域的名称以及分组**Group By**属性的值

为了了解分组是如何工作的，你可以单击**Add Data Field**按钮，从菜单中选择**Country**选项，在新创建的分组页眉区域增加一个位置。单击选中该字段，通过改变字体**Font**属性使字体更为醒目一些。

**使用代码添加分组和排序**

好的报表并不只是简单的展示数据，更重要的是将数据有效的组织起来。**C1Report**使用**groups**来实现数据的分组和排序。为了说明分组是如何工作的，我们将回到报表主题模板的代码创建过程中，通过国籍对员工进行分组。

下面的代码将展示如何创建一个分组对象，并且根据国籍对报表内容进行分组：

 Visual Basic

If chkGroup.Checked Then

' group employees by country, in ascending order

Dim grp As Group

grp = c1r.Groups.Add("GrpCountry", "Country", SortEnum.Ascending)

' format the Header section for the new group

With grp.SectionHeader

.Height = 500

.Visible = True

f = .Fields.Add("CtlCountry", "Country", 0, 0, c1r.Layout.Width, 500)

f.Calculated = True

f.Align = FieldAlignEnum.LeftMiddle

f.Font.Bold = True

f.Font.Size = 12

f.BorderStyle = BorderStyleEnum.Solid

f.BorderColor = Color.FromArgb(0, 0, 150)

f.BackStyle = BackStyleEnum.Opaque

f.BackColor = Color.FromArgb(150, 150, 220)

f.MarginLeft = 100

End With

' sort employees by first name within each country

c1r.Groups.Add("GrpName", "FirstName", SortEnum.Ascending)

End If

 C#

if (chkGroup.Checked)

{

// group employees by country, in ascending order

Group grp = c1r.Groups.Add("GrpCountry", "Country", SortEnum.Ascending);

// format the Header section for the new group

s = grp.SectionHeader;

s.Height = 500;

s.Visible = true;

f = s.Fields.Add("CtlCountry", "Country", 0, 0, c1r.Layout.Width, 500);

f.Calculated = true;

f.Align = FieldAlignEnum.LeftMiddle;

f.Font.Bold = true; **148**

f.Font.Size = 12;

f.BorderStyle = BorderStyleEnum.Solid;

f.BorderColor = Color.FromArgb(0, 0, 150);

//f.BackStyle = BackStyleEnum.Opaque;

f.BackColor = Color.Transparent;

f.BackColor = Color.FromArgb(150, 150, 220);

f.MarginLeft = 100;

// sort employees by first name within each country

c1r.Groups.Add("GrpName", "FirstName", SortEnum.Ascending);

}

每个分组都有页眉和页脚部分。在默认情况下，它们均不可见。但是上面的代码把页眉部分设为可见，是为了显示该分组所属的国籍。然后它使用**Country**字段新增了一个额外字段，并且将该字段的背景色设为纯色。

最后，代码中增加了另外一个分组，为了在国家分组中按照名字对员工排序。该分组仅用于排序，因此页眉和页脚部分都是不可见的。

完成设置之后，你需要调用**Render**方法呈现报表从而完成本例程。在**btnEmployees\_Click**的事件处理器中输入下面的代码：

 Visual Basic

' render the report into the PrintPreviewControl

ppv.Document = c1r.Document

 C#

// render the report into the PrintPreviewControl

ppv.Document = c1r.Document; :

下图为使用分组报表的实际效果图：

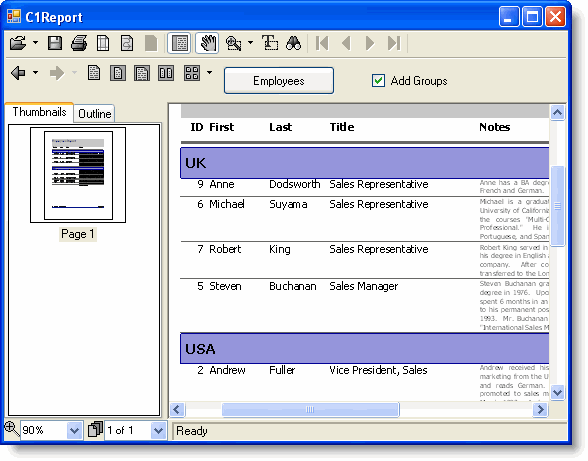


图 4

**数据排序**

你可以使用下面两种方式对数据进行排序：

1. 数据源对象自身排序（例如，使用SQL语句中的ORDER BY子句）

2. 在报表中添加分组并指定每个分组如何通过设置**Group和Sort**属性进行排序

使用**DataView.Sort**属性进行排序，你只需要使用一个列名列表（不需要列名表达式）。因此，如果你使用DatePart("yyyy", dateColumn)作为分组表达式，控件将自动根据**dateColunm**字段中的日期字段进行排序,而不是像你所想的那样使用日期中的年份。

根据日期排序，需要在数据表中增加一个计算列（通过修改SQL语句），然后根据计算列中的内容对报表进行分组或者排序。下面将为你展示一个XML的**Sort**属性，并举例说明这一过程。

下图为你展示的就是**C1ReportDesigner**中的**Sorting and Grouping**编辑器。你可以通过指定字段来实现分组排序：

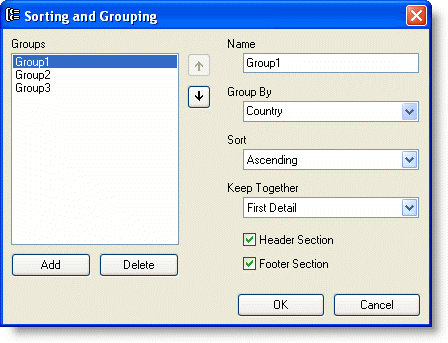


图 5

如果你同时使用上述两种方法，在报表中设置的Sort属性将具备更高的优先级（它应用于数据从数据库中检索之后）。

**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml文**件的"19: Sorting"章节，该文件保存在**ComponentOne**示例文件目录下。

**增加动态求和**

**C1Report**字段对象有一个**RunningSum**属性，从而使其很容易的对分组或者整个报表中的数据进行动态求和。

**在分组中增加动态求和**

在分组中进行动态求和，需要完成以下步骤：

1. 打开**C1ReportDesigner**应用程序。更多如何使用**C1ReportDesigner**的信息，可以参阅葡萄城系列文档中的Accessing C1ReportDesigner from Visual Studio章节。（《Reports for .NET DesignerEdition》238页）

2. 创建一个新报表或者打开一个已存在的报表。通过**C1ReportDesigner**打开之后，你就可以对报表属性进行修改。

3. 单击**Design**按钮，开始编辑报表。

4. 在设计模式下，从属性窗口上方的下拉列表中选择报表。报表中的有效属性将显示在这里。

5. 在报表中增加一个calculate字段：

在**Designer**的工具栏中，单击**Add Calculated Field**按钮。

在VBScript编辑器中,输入以下脚本:

Sum(ProductSalesCtl)

将鼠标拖过报表分组的页眉区域，然后光标会变成十字线样式。单击并拖动鼠标来重新定义该字段所占区域，然后释放鼠标按键完成新字段的创建。

6. 将**RunningSum**属性设置为**SumOverGroup**（注意，如果想要显示该属性，属性过滤必须关闭。属性过滤设置按钮是属性窗口上的漏斗图标）

**在整个报表中增加动态求和**

如果想要实现跨页面动态求和，你需要使用脚本来完成这一功能。例如，你可以增加一个**pageSum**字段到报表中，使用脚本对其进行更新。完成以下步骤来实现该功能：

1. 打开**C1ReportDesigner**应用程序。更多如何使用**C1ReportDesigner**的细节，可以参阅葡萄城系列文档中的Accessing C1ReportDesigner from Visual Studio章节。（《Reports for .NET DesignerEdition》238页）

2. 创建一个新报表或打开一个已存在的报表。通过**C1ReportDesigner**打开之后，你可以对报表属性进行修改。

3. 单击**Design**按钮，开始编辑报表。

4. 在设计模式下，从属性窗口上方的下拉列表中选中相应报表。这里将显示报表中的可用属性。

5. 找到**OnPage**属性，单击旁边的空白字段，然后点击**省略号**按钮。

6. 在**VBScript**编辑框中，输入下面的VB表达式脚本代码：

' VBScript: Report.OnPage

pageSum = 0

7. 在属性窗口上方的下拉列表中选中**Detail**选项，此处将显示**Detail**区域的可用属性。

8. 找到**OnPrint**属性，单击旁边的空白字段。然后单击**省略号**按钮。

9. 在弹出VBScript编辑器中输入以下VBScript表达式脚本代码:

' VBScript: Detail.OnPrint

pageSum = pageSum + UnitsInStock

**示例报表**

完整的报表可参阅在报表模板文件下**CommonTasks.xml文**件的"17: Running Sums"章节，该文件保存在**ComponentOne**示例文件目录下。

**添加汇总和其他合计**

C1Report支持合计表达式的所有计算字段。合计表达式包含所有常用的函数如**Sum**, **Avg**, **Min**, **Max**, **Count**, **Range**, **StDev**等等。

所有的合计功能都会使用一个表达式作为参数，并且根据表达式在报表中所处的位置来决定其计算范围。例如，在分组页眉和或者页脚中的合计，其范围是在分组内。在报表的页眉或者页脚中的合计，其范围是在报表内。

例如，下面的合计表达式将返回范围（分组或报表）内的Sales字段值的总和。

Sum(Sales)

下面的合计表达式将会返回报表中支付销售税款的总量（假设销售税为8.5%）：

Sum(Sales \* 0.085)

你可以使用域（domain）作为合计表达式的第二参数，从而缩小其作用范围。域参数是一个表达式，它将决定合计表达式当前范围内的值是否应该被包含在本次合计计算中。

例如，下面的合计表达式将返回所有产品类别为1的Sale字段值的总和：

Sum(Sales, Category = 1)

下面的合计表达式将返回金额超过10000美元的数量。

Count(\*, Sales > 10000)

**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml文**件的"13: Subtotals and other Aggregates"章节，该文件保存在**ComponentOne**示例文件目录下。

**建立交叉报表**

交叉报表会在两个维度上对数据进行分组（横向或纵向）。交叉报表主要用于汇总报表中大量交叉引用的数据信息。

想要创建交叉报表，首先你需要调用GROUP BY查询语句将数据汇总到行中，然后使用转化(某个支点)服务创建分组的列。转化服务通常由数据库服务自身提供，它可以是一个自定义程序，或者你也可以使用**C1Report**内置的域合计功能。

无论何时，交叉报表中最重要的元素都是数据的原始汇总视图。例如，一个典型的汇总视图如下所示：

|  |  |  |
| --- | --- | --- |
| **Year** | **Quarter** | **Amount** |
| 1990 | 1 | 1.1 |
| 1990 | 2 | 1.2 |
| 1990 | 3 | 1.3 |
| 1990 | 4 | 1.4 |
| 1991 | 1 | 2.1 |
| 1991 | 2 | 2.2 |
| 1991 | 3 | 2.3 |
| 1991 | 4 | 2.4 |

我们首先为每季度添加一个新列，并且将合并之后的值加入到新列中。通过这种方式改变数据后的效果如下图所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **Total** | **Q1** | **Q2** | **Q3** | **Q4** |
| 1990 | 5 | 1.1 | 1.2 | 1.3 | 1.4 |
| 1991 | 9 | 2.1 | 2.2 | 2.3 | 2.4 |

您可以使用**C1Report**合计函数完成该操作。该报表将会依据年份进行分组。Detail区域将会隐藏，分组的页眉将会包含的合计表达式如下所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Year** | **Total** | **Q1** | **Q2** | **Q3** | **Q4** |
| [Year] | Sum(Amount) | Sum(Amount, Quarter=1) | Sum(Amount, Quarter=2) | Sum(Amount, Quarter=3) | Sum(Amount, Quarter=4) |

第一个合计表达式将计算当前年份中的销售总额，而指定季度的合计将会通过指定域的方式来限制合计表达式只能获取指定月份的数值。