**RDLC使用总结**

1. 为什么要使用RDLC报表（简介）

VS .NET开发中，用什么进行报表设计？可能的回答是Crystal Report，当然，必须承认Crystal Report的功能还是非常强大的，被Business Object收购以后，商业的成分也在逐渐增加，也形成了一定规模的用户群。

Visual Studio .NET进入2005版本以后，Crystal Report与IDE的结合更“紧密”了，至少我们看不到那个讨厌的注册窗口了。但是，Microsoft似乎并不容忍在自己的超级工具中竟然没有报表工具，于是Report Viewer Control出现了，我把它的报表称为RDLC报表。

在VS .NET 2005之前，SQL Server Reporting Services中已经提供了一种被称为报表定义语言（Report Definition Language, RDL）的语言；在VS .NET 2005中，Microsoft提供了针对这种报表的设计器，并提供了在WinForm和WebForm中使用这种报表的能力。Microsoft将这种报表的后缀定为RDLC，RDL仍然是Report Definition Language的缩写，那么C代表什么呢？C代表Client-side processing，凸显了它的客户端处理能力。

这种报表的易用性和可定制性让我们完全有理由放弃Crystal Report，让我们来看看它的强大功能吧：

* 1. 简单易用的控件，特别是Table控件，非常方便字段在报表上的排列；
  2. 灵活的可定制性，用XML来描述一个报表相关的一切，不用说了吧；
  3. 高度可编程性，在你的项目中，甚至不需要有一个报表文件，通过代码就可以实现报表生成、预览和打印等一系列操作；
  4. 支持DrillThrough数据钻取功能；
  5. 导出的Excel文件格式非常完美，而且并不需要安装Excel；
  6. 数据源处理极其方便，开发人员可以自己接管数据库连接、取数，然后将数据结果赋值给RDLC的数据集即可。
  7. 展示和数据分离，程序员更是可以编写一个RDLC设计器交有用户使用，这样用户就可以自行设计RDLC报表格式，程序员只负责定制数据接口即可。

RDLC MSDN：<http://msdn.microsoft.com/zh-cn/library/ms251671(VS.80).aspx>

1. RDLC简单Demo

本文所有RDLC例子都是基于VS2005 C#，所有ReportViewer的处理模式都是Local模式。本报表Demo显示上文中提到的RDLC手册各节内容，下述为Demo的详细steps。

1. 新建项目，选择Windows应用程序项目类型，输入工程名称RDLCDev；
2. 修改Form1窗体名称为FrmRdlcUserGuide，在工具箱—>数据Tab选项卡中选择ReportViewer控件，将其拖入到FrmRdlcUserGuide，默认命名为reportViewer1，调整ReportViewer控件的大小和布局；
3. 在解决方案资源管理器中选择RDLCDev工程，鼠标右击RDLCDev工程，添加新建项，在添加新项模板中选择“报表”，取名为rdlcuserguide.rdlc，单击添加按钮，然后在解决方案资源管理器中RDLCDev工程下会增加一个rdlcuderguide.rdlc文件，并且会自动打开RDLC设计器；
4. 在RDLC设计器中添加11个TextBox控件，1个作为标题，其余10个作为RDLC使用手册具体内容，对每个TextBox控件进行TextAlign、Color、BorderStyle、Font和TextDecoration等相关属性设置，设计完毕之后，大致如下图所示。



1. 在FrmRdlcUserGuide窗体Load事件中添加下述代码

private void Form1\_Load(object sender, EventArgs e)

{

this.reportViewer1.ProcessingMode = Microsoft.Reporting.WinForms.ProcessingMode.Local;

this.reportViewer1.LocalReport.ReportPath = "rdlcuserguide.rdlc";

this.reportViewer1.RefreshReport();

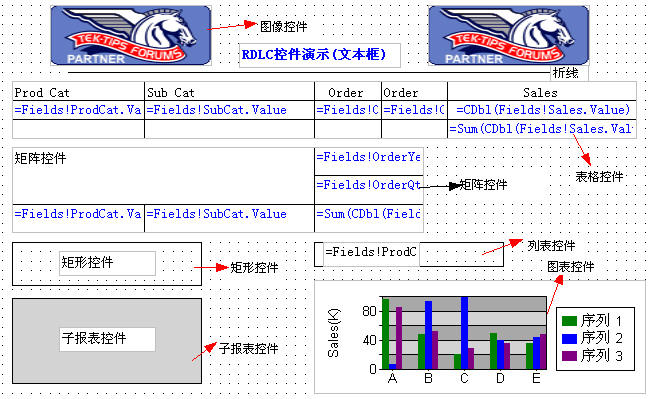
}

1. 编译、运行RDLCDev工程，运行结果大致如下图所示。



1. RDLC常用控件介绍

RDLC设计工具箱上总共有9种控件：文本框(TextBox)、折线(Line)、表(Table)、矩阵(Matrix)、矩形(Rectangle)、列表(List)、子报表(SubReport)、图表控件(Chart)、图像控件(Image)。下图ctlsDemo.rdlc设计器所示为所有控件的设计演示。



1. 由于表格控件、矩阵控件、列表控件和图表控件必须填充数据集，因此把所有控件布局完毕之后，创建一个数据集。解决方案资源管理器中，选中RDLCDev工程，右击添加新建项，在新建项类型模板里选择“数据集”，取名“ctlsDemo.xsd”,单击“添加”。

ctlsDemo.xsd添加完毕之后，在解决方案资源管理器中会增加一个ctlsDemo.xsd文件，并前VS IDE会自动打开ctlsDemo.xsd设计页面。在设计页面中右击，添加Datatable，将其命名为vSales，该表描述的是产品的销售额信息，为vSales表添加列，列信息如下：

ProdCat System.String

SubCat System.String

OrderYear System.Int32

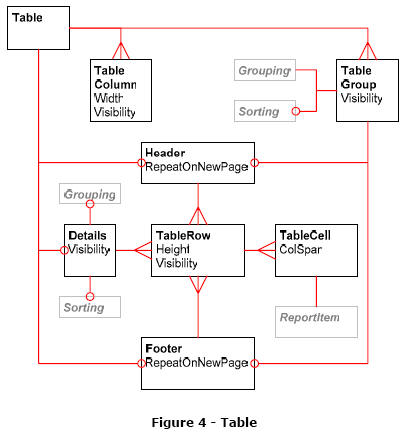
OrderQtr System.String

Sales System.Double

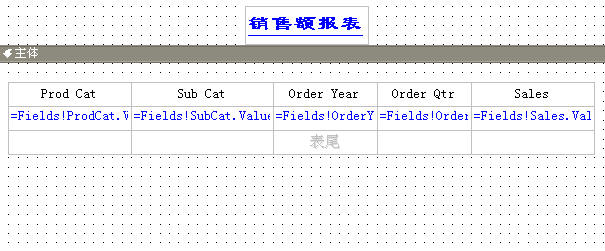
1. 文本框控件、折线控件、矩形控件使用相对简单，这里不细说；
2. Image控件，现在RDLC设计器模式下，选择“报表”菜单，“嵌入图像”，在“嵌入图像”对话框中，载入一个本地图像文件即可。然后设置Image控件的Source属性为Embedded，设置Value属性为刚才处理的嵌入图像即可。当然还有别的方式可以显示图像，大家可以Google一下，关键字为RDLC、嵌入图像；
3. 图表控件，使用图表控件的一个关键是要设置好图表属性页面中的数据TAB页相关内容。主要有值、类别组和序列组定义，大家可以参考Demo；
4. 表格控件、矩阵控件和子报表控件，在后续内容中会详细介绍；

4.1 表格控件使用介绍

* 1. Table控件是RDLC报表显示数据的一个核心控件，关于Table控件，在微软提供的RDLC规范里有详细的介绍，其大致内容可以用下述的一张类UML图来表示；



* 1. 新建报表文件，命名为ctlTableDemo.rdlc；
  2. 在ctlTableDemo.rdlc设计器中，拖入Table控件，默认情况下是3行3列，其中第1行为表头行（我们理解为列标题行），第二行为详细信息行（我们理解为报表内容展示区域），第三行为表尾行（我们理解为汇总区域）。可以整行选中，然后右击鼠标添加行或者删除行，也可以整列选中，然后右击鼠标添加或者删除列。上文中提到的数据集合有5个字段，因此需要5列来显示，为此需要增加两列。
  3. 在数据源窗口中（如果没有显示的话，选择【数据】菜单下的【显示数据源】即可）选择上文中创建的ctlsDemo.xsd数据集，将ProdCat、SubCat、OrderYear 、OrderQtr和Sales分别拖入到Table控件的详细信息行的不同列中去。
  4. 设置标题行文字显示都居中，Sales的内容显示右对齐，ProdCat、SubCat、OrderYear、OrderQtr和Sales文本框的字体颜色全部设置成Blue，Sales文本框由于显示的内容为销售额，因此需设置数字显示格式，小数位数。鼠标右击Sales文本框，选择【属性】，在文本框属性窗口中，选择格式Tab页，在格式代码处，选择数字1，234.00格式即可，选择完毕之后该文本框的数字就以逗号千分位作为分割符，保留2位小数显示，效果如下图所示。



* 1. 插入一个组实现分组统计功能。选中整行，鼠标右击，选择【插入组】，弹出分组和排序属性对话框，设置分组名称、分组方式、是否显示组头或者组尾等属性即可，本文分组设置如下：

·分组名称：table1\_Group1

·分组表达式：=Fields!ProdCat.Value；=Fields!SubCat.Value；=Fields!OrderYear.Value，实现产品年度销售额汇总统计功能。

·不显示组头、显示组尾（至于组头、组尾是个什么样的东西，大家只要动手试一下就一目了然了）

·在textbox15中输入汉字“合计”，在textbox16文本框中输入合计表达式=Sum(CDbl(Fields!Sales.Value))，设置完之后效果如下图所示。



* 1. 新建窗体FrmCtlTableDemo，然后在该窗体上添加ReportViewer控件，在窗体FrmCtlTableDemo代码窗口里输入以下代码：

private void FrmCtlTableDemo\_Load(object sender, EventArgs e)

{

this.reportViewer1.ProcessingMode = Microsoft.Reporting.WinForms.ProcessingMode.Local;

this.reportViewer1.LocalReport.ReportPath = @"rdlc\ctlTableDemo.rdlc";

//

reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("ctlsDemo\_vSales", LoadData()));

//

this.reportViewer1.RefreshReport();

}

private DataTable LoadData()

{

DataSet dataSet = new DataSet();

dataSet.ReadXml(@"data\ctlsDemo.xml");

return dataSet.Tables[0];

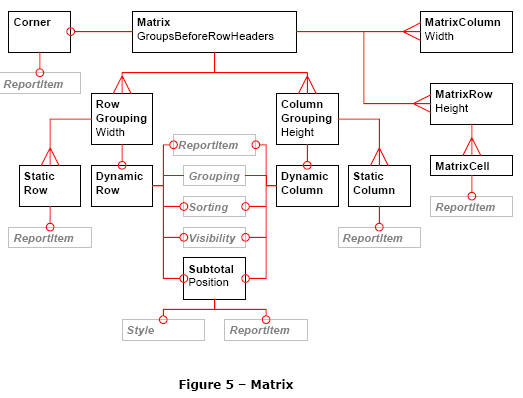
}

* 1. 运行效果如下图所示



4.2 矩阵控件使用介绍

* + 1. 矩阵控件主要用于显示CrossTable之类的报表，因此其列数可以是Static的，也可以是Dynamic并且Can grow的，关于矩阵控件，在RDLC协议规范里也有介绍，其大致内容可以用下图表示。



* + 1. 新建报表文件，命名为ctlMatrixDemo.rdlc；
    2. 在ctlMatrixDemo.rdlc设计器中，拖入Matrix控件，默认情况下是2行2列，左上角单元格为空白区域（通常用来显示报表摘要信息如产品/年度等），右上单元格显示为“列”，单击此处可以增加列分组，左下单元格显示为“行”，单击此处可以添加行分组，右下单元格显示为“数据”，单击此处可以添加静态行、静态列。
    3. 在矩阵左上角单元格空白区域拖入矩形控件，然后在该矩形控件中拖入一个折线控件和两个文本框控件，折线当做矩形的对角线，两个文本框内容为产品和年度分布在折线两侧。
    4. 增加两个行分组，分别为matrix1\_RowGroup1和matrix1\_RowGroup2，matrix1\_RowGroup1的表达式为=Fields!ProdCat.Value，matrix1\_RowGroup2的表达式为=Fields!SubCat.Value。
    5. 增加两个列分组，分别为matrix1\_ColumnGroup1何matrix1\_ColumnGroup2，matrix1\_ColumnGroup1的表达式为=Fields!OrderYear.Value，matrix1\_ColumnGroup2的表达式为=Fields!OrderQtr.Value。
    6. 在“数据”单元格中编辑表达式=Sum(CDbl(Fields!Sales.Value))
    7. 设计完之后，效果如下图所示；



* + 1. 新建窗体FrmCtlMatrixDemo，然后在该窗体上添加ReportViewer控件，在窗体FrmCtlMatrixDemo代码窗口里输入以下代码：

private void FrmCtlMatrixDemo\_Load(object sender, EventArgs e)

{

this.reportViewer1.ProcessingMode = Microsoft.Reporting.WinForms.ProcessingMode.Local;

this.reportViewer1.LocalReport.ReportPath = @"rdlc\ctlMatrixDemo.rdlc";

//

reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("ctlsDemo\_vSales", LoadData()));

//

this.reportViewer1.RefreshReport();

}

private DataTable LoadData()

{

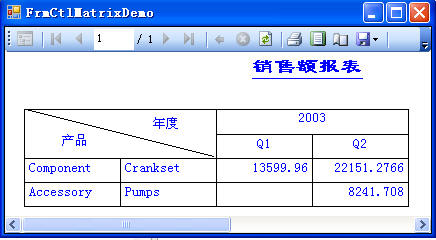
DataSet dataSet = new DataSet();

dataSet.ReadXml(@"data\ctlsDemo.xml");

return dataSet.Tables[0];

}

* + 1. 运行效果如下图所示



4.3 子报表控件使用介绍

子报表作为主报表的一个补充，在显示明细内容的时候比较管用，本文从如何填充子报表数据，如何传递参数两方面介绍子报表如何使用。

1. 新建RDLC报表，命名为subReportDemo.rdlc；
2. 在subReportDemo.rdlc设计器中，拖入一个文本框和一个子报表控件；
3. 设置文本框的文本显示为“子报表控件演示”；
4. 设置子报表控件属性，主要设置其“子报表”的值，这里设为上文讲到的ctlTableDemo报表，效果如下图所示；



1. 新建一个窗体FrmSubReportDemo，添加如下代码

private void FrmCtlSubReportDemo\_Load(object sender, EventArgs e)

{

reportViewer1.LocalReport.SubreportProcessing += new SubreportProcessingEventHandler(LocalReport\_SubreportProcessing);

//

this.reportViewer1.ProcessingMode = Microsoft.Reporting.WinForms.ProcessingMode.Local;

this.reportViewer1.LocalReport.ReportPath = @"rdlc\subReportDemo.rdlc";

//

reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("ctlsDemo\_vSales", LoadData()));

//

this.reportViewer1.RefreshReport();

}

void LocalReport\_SubreportProcessing(object sender, SubreportProcessingEventArgs e)

{

e.DataSources.Add(new ReportDataSource("ctlsDemo\_vSales", LoadData()));

}

private DataTable LoadData()

{

DataSet dataSet = new DataSet();

dataSet.ReadXml(@"data\ctlsDemo.xml");

return dataSet.Tables[0];

}

这里，区别于其它显示RDLC报表程序的代码就是对子报表的处理，本文在FrmCtlSubReportDemo\_Load事件里添加了一个对子报表事件的处理：reportViewer1.LocalReport.SubreportProcessing += new SubreportProcessingEventHandler(LocalReport\_SubreportProcessing);

1. 关于子报表参数如何传递，在RDLC报表参数介绍一文中介绍；
2. 运行效果如下图所示



1. RDLC参数介绍

5.1 报表参数处理

* + 1. 修改上文提高的ctlTableDemo.rdlc报表为例，该报表用于显示产品每季度的销售额。修改该报表，给该报表添加一个参数prod\_cat，用于过滤只显示产品系统为参数prod\_cat值对应的产品季度销售额。
    2. 在ctlTableDemo.rdlc设计器中，选择【报表】菜单，【报表参数】菜单，在报表参数界面中，添加一个报表参数取名为“prod\_cat\_sub”，类型为String，提示信息为“输入产品系列名称”，允许空白值；
    3. 选中table1控件，鼠标右击，选择【属性】，选中【筛选器】Tab页，在筛选器列表中添加：=Fields!ProdCat.Value = =Parameters! prod\_cat\_sub.Value；
    4. 在FrmCtlTableDemo窗体中添加下述代码，用红色表示的2行代码：

private void FrmCtlTableDemo\_Load(object sender, EventArgs e)

{

this.reportViewer1.ProcessingMode = Microsoft.Reporting.WinForms.ProcessingMode.Local;

this.reportViewer1.LocalReport.ReportPath = @"rdlc\ctlTableDemo.rdlc";

//

reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("ctlsDemo\_vSales", LoadData()));

//

ReportParameter p = new ReportParameter("prod\_cat\_sub", "Component");

reportViewer1.LocalReport.SetParameters(new ReportParameter[] { p });

//

this.reportViewer1.RefreshReport();

}

* + 1. 运行结果如下图所示，显示Component产品系列的产品季度销售额；



5.2 子报表参数处理

1. 在程序里面，本人还未找到直接向子报表传递参数的方法，但是可以用其它的方法变通实现，在主报表的subreport控件属性里面设置参数，通过主报表传递过去。
2. 以上文提到的FrmSubReportDemo为例，其主报表为subReportDemo.rdlc，先设置主报表的参数。在subReportDemo.rdlc报表设计器中选择【报表】，【报表参数】，在报表参数对话框中，增加一个参数。参数名称为prod\_cat，参数类型为String，参数提示为“输入产品系列名称”。
3. 选中子报表subreport1，鼠标右击，选择【属性】，选择【参数】tab页，添加一个参数。参数名称为“prod\_cat\_sub”，参数值=Parameters!prod\_cat.Value。为此，主报表的参数名与子报表的参数名称要不一致。
4. 打开subreport1对应的子报表设计器，本文为ctlTableDemo.rdlc，该ctlTableDemo.rdlc的参数在上文中已经设置，这里不再赘述。
5. 运行效果如下图所示。(源码网整理：[www.codepub.com](http://www.codepub.com/))



1. RDLC钻取功能实现

本Demo在主报表会显示一个部门列表（部门ID、部门名称等），然后通过钻取功能可实现特定部门的员工信息展示，下述为具体操作步骤。

* + 1. 准备好两个数据源，本文为2个xml文件（departments.xml和Employees.xml），这两个文件放在项目的Data文件夹下，特别注意的是这两个文件的【复制到输出目录】属性要选择“始终复制”，负责工程编译的时候不会将这两个文件生成到bin目录下。
    2. 新建报表，命名为“drillthrough.rdlc”；新建一个数据集，命名为“departments”，在该数据集中添加一个Datatable，表结构如下：

DepartmentID Int32

Name String

GroupName String

ModifiedDate DateTime

rowguid String

* + 1. 在drillthrough.rdlc设计器中，拖入一个文本框和一个Table控件。文本框显示内容为“Drill Through Demo”，在Table控件中拖入DepartmentID和Name两列。
    2. 添加Name列的导航、钻取功能。右击Name文本框，选择【属性】，选择【导航】Tab页，在“超级链接”处选择“跳至报表”（该报表为显示钻取实现的目标报表，假设为drillthroughDetail.rdlc，且已经存在，因此在实践时，可以再本操作之前，先创建一个drillthroughDetail.rdlc报表），设置“跳至报表”为drillthroughDetail，点击【参数】按钮，会弹出参数设置对话框。在对话框中，添加一个参数，名称为DepartmentID，值为=Fields!DepartmentID.Value；
    3. 设计drillthroughDetail.rdlc报表，先新建一个该报表对应的数据集，名称为employees，添加一个DataTable，名称为employee，employee表部分内容如下：

<table>

<EmployeeID>1</EmployeeID>

<FirstName>Terri</FirstName>

<AddressID>1</AddressID>

<ShiftID>1</ShiftID>

<LastName>Duffy</LastName>

<MiddleName>Lee</MiddleName>

<NameStyle>false</NameStyle>

<NationalIDNumber>245797967</NationalIDNumber>

<Title>VP Engineering</Title>

<HireDate>1998-03-03T00:00:00.0000000-08:00</HireDate>

<BirthDate>1961-09-01T00:00:00.0000000-07:00</BirthDate>

<LoginID>adventure-works\Terri</LoginID>

<EmailAddress>Terri@adventure-works.com</EmailAddress>

<DepartmentID>1</DepartmentID>

<MaritalStatus>S</MaritalStatus>

<EmergencyContactName>Tad Orman</EmergencyContactName>

<EmergencyContactPhone>586-883-8338</EmergencyContactPhone>

<SalariedFlag>true</SalariedFlag>

<Gender>F</Gender>

<ManagerID>300</ManagerID>

<PayFrequency>2</PayFrequency>

<BaseRate>63.4615</BaseRate>

<VacationHours>1</VacationHours>

<SickLeaveHours>20</SickLeaveHours>

<SalesPersonFlag>false</SalesPersonFlag>

<CurrentFlag>true</CurrentFlag>

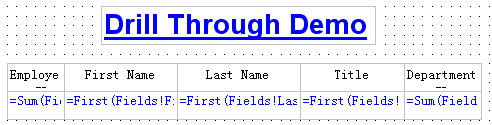
<ModifiedDate>2003-01-15T19:26:13.9000000-08:00</ModifiedDate>

<rowguid>1b76e019-0d60-4d48-b405-c9144fb9b3ab</rowguid>

</table>

从表的内容中，我们可以大致判断出其结构是什么样的啦。

* + 1. 拖入一个文本框控件和表格控件到drillthroughDetail.rdlc报表设计器中，并从employees数据集中拖入相应字段到表格控件中，设置完毕之后，效果如下图所示。



* + 1. 设置drillthroughDetail.rdlc报表的参数：参数名称为DepartmentID，类型为Integer；
    2. 设置drillthroughDetail.rdlc报表表格控件的筛选器，表达式为=CInt(Fields!DepartmentID.Value)，运算符为=，值为=Parameters!DepartmentID.Value。
    3. 新建一个窗体FrmDrillThrough，添加一个ReportViewer控件，在代码编辑窗口，添加下述代码：

private void FrmDrillThrough\_Load(object sender, EventArgs e)

{

reportViewer1.ProcessingMode = ProcessingMode.Local;

reportViewer1.LocalReport.ReportPath = @"Rdlc/drillthrough.rdlc";

reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("departments\_department", LoadData(@"Data\departments.xml")));

reportViewer1.RefreshReport();

}

//读取XML数据

private DataTable LoadData(string xmlFilePath)

{

DataSet dataSet = new DataSet();

dataSet.ReadXml(xmlFilePath);

return dataSet.Tables[0];

}

private void reportViewer1\_Drillthrough(object sender, DrillthroughEventArgs e)

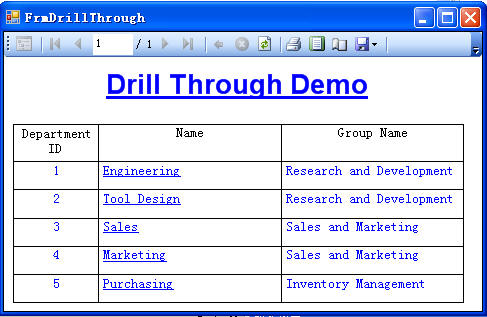
{

LocalReport localReport = (LocalReport)e.Report;

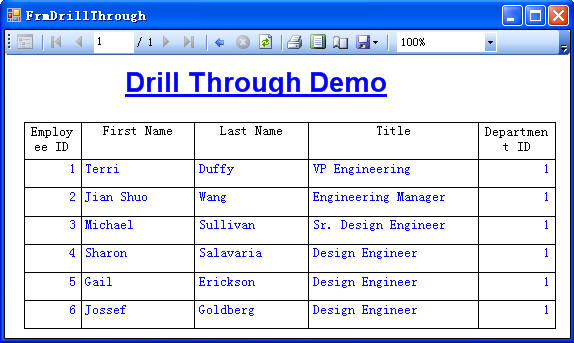
localReport.DataSources.Add(new ReportDataSource("employees\_employee", LoadData(@"Data\Employees.xml")));

}

* + 1. 运行效果如下图所示。



鼠标移到Name一列，会变成手的形状，如单击Engineering，就会钻取取来Engineering部门的员工信息，如下图所示。



1. RDLC自定义工具条
   * + 1. 自定义工具条可以理解为对ReportViewer的ToolBar功能的一种变通实现，如添加代码实现ToolBar刷新按钮的功能、ToolBar页导航功能。

如刷新功能代码实现为：

this.rptViewer.RefreshReport();

如ToolBar停止按钮的代码实现为：

this.rptViewer.CancelRendering(0);

。。。

* + - 1. 在做项目的时候，如果用ReportViewer控件做报表展示工具的话，打印和导出的代码实现，应该要掌握。

导出功能：ReportViewer导出都是通过Render来实现的，如下代码为导出Excel

Microsoft.Reporting.WinForms.Warning[] Warnings;

string[] strStreamIds;

string strMimeType;

string strEncoding;

string strFileNameExtension;

//this.rptViewer.LocalReport.Render(

byte[] bytes = this.rptViewer.LocalReport.Render("Excel", null, out strMimeType, out strEncoding, out strFileNameExtension, out strStreamIds, out Warnings);

string strFilePath = @"D:\exportdemo.xls";

using (System.IO.FileStream fs = new FileStream(strFilePath, FileMode.Create))

{

fs.Write(bytes, 0, bytes.Length);

}

打印例子下载<http://files.cnblogs.com/waxdoll/RDLCPrint.rar>，打印也是一个间接行为，有点类似ReportViewer控件需要通过点击打印按钮来打印一样，并没有提供直接的打印函数，为此本人封装了打印功能(在附件的whhrdlc.dll文件中)并开放了打印函数，用户可以调用下面语句打印报表：

//打印出错时，错误信息描述

String errMsg = String.Emptys;

//调用打印函数(弹出打印设置对话框)返回false打印失败；

bool bln = Printer.PrintDialog(this.reportViewer1.LocalReport, out errMsg);

//调用打印函数(直接打印)返回false打印失败；

bool bln = Printer.Print (this.reportViewer1.LocalReport, out errMsg);

1. 设计器实现

8.1 RDLC XML分析

* + - 1. RDLC文件其实就是一个XML文件，有很多结点和元素，在解决方案资源管理器中选中某个rdlc文件，鼠标右击，选择【打开方式】，在打开方式对话框中选择XML编辑器，单击确定按钮即可。

在xml文件中我们可以看到DataSources、PageHeader、DataSets、ReportItems等结点，网上有篇Vs2005报表设计器XML分析总结的博文

<http://blog.csdn.net/sutter/archive/2008/01/14/2044253.aspx>，写的比较详细，大家可以参考。

* + - 1. 这里比较关键的就是DataSet的处理，Rdlc报表无论是表格显示还是矩阵显示或者列表显示数据，都需要关联DataSet。在前面提到的所有篇幅中，都是先创建一个数据集，并为该数据集添加一个DataTable，然后将表字段拖放到表格控件、矩阵控件或者列表控件，用以显示数据。这么做，从效率上来看不是最高的，因为手工产生一个目标DataTable比较耗时。为此，我们可以先直接编辑rdlc的xml文件，编辑器dataset的内容即可，如下所示：

<DataSets>

<DataSet Name="ctlsDemo\_vSales">

<Fields>

<Field Name="ProdCat">

<DataField>ProdCat</DataField>

<rd:TypeName>System.String</rd:TypeName>

</Field>

<Field Name="SubCat">

<DataField>SubCat</DataField>

<rd:TypeName>System.String</rd:TypeName>

</Field>

<Field Name="OrderYear">

<DataField>OrderYear</DataField>

<rd:TypeName>System.Int32</rd:TypeName>

</Field>

<Field Name="OrderQtr">

<DataField>OrderQtr</DataField>

<rd:TypeName>System.String</rd:TypeName>

</Field>

<Field Name="Sales">

<DataField>Sales</DataField>

<rd:TypeName>System.Double</rd:TypeName>

</Field>

</Fields>

<Query>

<DataSourceName>DummyDataSource</DataSourceName>

<CommandText />

<rd:UseGenericDesigner>true</rd:UseGenericDesigner>

</Query>

<rd:DataSetInfo>

<rd:DataSetName>ctlsDemo</rd:DataSetName>

<rd:TableName>vSales</rd:TableName>

</rd:DataSetInfo>

</DataSet>

</DataSets>

修改其Field定义，修改Dataset的Name即可，最后别忘了保存即可。

8.2 RDLC设计器实现

1. 在8.1中已经提到了RDLC报表文件，其实就是一个xml文件，那么只要我们用程序动态产生一个xml文件，只要这个xml文件的节点定义、元素定义、属性定义符合RDLC协议规范即可。
2. 关于RDLC设计器，已经有人可以很好地实现它了，而且是OpenSource的，大家可以去看看。

因为不知道如何在博文中上传附件，把相关资料已经上传到CSDN，大家可以搜索“RDLC使用手册”下载。

### 使用RDLC报表(一)

1 建立数据源

启动VS2005新建一个窗体项目，命名为TestProj

在左边的窗体内选择“添加新数据源”或在菜单上操作“添加新数据源”：

选择后出现对话窗体，选择“数据库”，进入“下一步”：

本数据源以SqlServer2000所提供的Northwind数据库为例，因此在数据库内选择驱动程序为SqlServer，具体操作视所用数据库而定（如果连接为Oracle，建议用Oracle所提供的ODP for .NET，地址：

http://www.oracle.com/technology/global/cn/software/tech/windows/odpnet/index.html）。

选择数据源后，指定数据库文件，并进行测试，如果连接不通，请检查你的软件设置。

进行下一步，至显示出库文件内的表及视图，然后打开表，并选择其中的Employees，选择完成。

在数据源窗口中显示出所选择表及表内的字段，以备使用。

同时在资源管理器中，也可以看到新增加的数据文件。

2 报表浏览器

在新建的窗体内，放入报表浏览控件：

此报表浏览控件集成了报表查看及打印预览的功能，可直接输出至打印机，也可直接输出至Excel文件或PDF文件，对于喜欢看Excel表格的客户来说，这个报表控件非常不错。关于此报表的来源及台前幕后的各种说法，请自己查询相关的网站。

3 建立报表文件

选中报表浏览器控件后，在控件的右上角会出现一个小三角，点击后，出现一个菜单，选择“设计新报表”或在解决方案资源管理器中添加一个新的报表文件：

双击“报表项”内的“表”，则设计空白表内生成一个表格，我们可以在表格内添加数据项。

此报表的设计方式类似于MS以前的VB下报表设计环境。在报表设计器中，我将一份报表分为五个部分，从上至下为：报表标题区、列标题区、数据显示区、合计及页脚。如果将报表的设计形式显示出页眉及页脚，报表的形式会看得更清楚些：

选择页眉及页脚：

显示结果：

现在开始设计报表：

在“页眉”区内放入一个文本框，将内容更改为“测试报表”，居中，字体放大加粗：

从数据源窗口中选择合适的列，用鼠标拖到报表内的“详细信息”表格上，数据会自动填入：

将设计的报表保存后，返回报表浏览窗口，并为报表浏览器选择我们设计的报表：

选中报表文件后，菜单也有所改变：

再为此报表文件指定数据源，选择数据源：

确定，保存文件并运行测试文件。

4 预览

报表生成后预览如下：

是不是很简单？我们也可以对这个简单的报表进行美化，当然，我加上美化的一些功能后，可能效果更差，但此处只是为了演示用法。

Ø 为数据加表格。

在报表设计器中，选择需要加表格框的区域

然后在工具栏中选择边框工具

加什么样的框线根据需要，此处我选择“所有框线”。

看上去比原来的灰线能稍黑一些，在预览时就能看出来了。

Ø 加打印时间显示

加入一个显示打印时间的文本框，在上面按右键，选属性：

在“文件框属性”内，选择“值”后面的表达式生成：

在表达式中选择时间：

不要忘记表达式的前面一定要有一个“＝”，如果没有，手工加上一个。为了更符合我们的习惯，前面可以加个说明，如打印时间等，表达式的写法就要稍做改变：还是以“＝”开头，后面加上“打印时间：”，再以“＆”连接生成的Now函数即可得到我们想要的样子。

Ø 为行加序号

为了演示，我将序号列加在了表格了最后列

加入一个新的列：

在显示序号的列中输入相关的值：=RowNumber(Nothing)

在此表达式的后面是=RowNumber(Nothing)，不是null，不要搞错。

三项功能已经完成，预览：

### 使用RDLC报表(二)

1.開發前要先安裝MicroSoft ReportViewer 9.0版(原VS2005的ReportViewer版本為8.0)，8.0版本不支持本地打印[Download](http://www.microsoft.com/downloads/details.aspx?FamilyID=cc96c246-61e5-4d9e-bb5f-416d75a1b9ef&DisplayLang=en)

2.Web.Config 須加入以下設定

<add assembly="Microsoft.ReportViewer.WebForms, Version=9.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A" />  
<add assembly="Microsoft.ReportViewer.Common, Version=9.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A" />

<httpHandlers>  
      <add verb="\*" path="Reserved.ReportViewerWebControl.axd" type = "Microsoft.Reporting.WebForms.HttpHandler, Microsoft.ReportViewer.WebForms, Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />  
</httpHandlers>

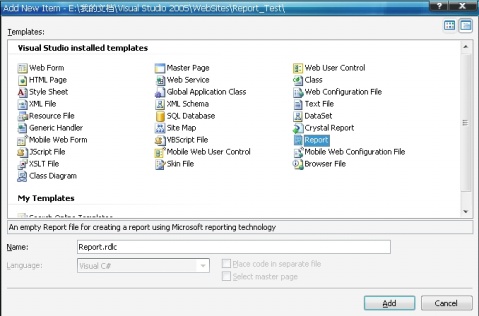
### 使用RDLC报表(三)

最近正在做微软的RDLC报表（即ReportView），把自己的经验写了来，下面开始做一张分组统计的简单报表吧~



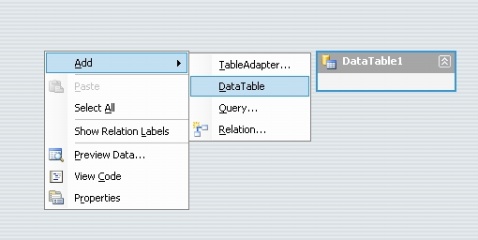
（最终图）

   一、先建一张报表：

[](http://img.blog.163.com/photo/HbCahC-Eyly380VvU3hEtQ==/5150429123852152153.jpg)

二、设计dataset

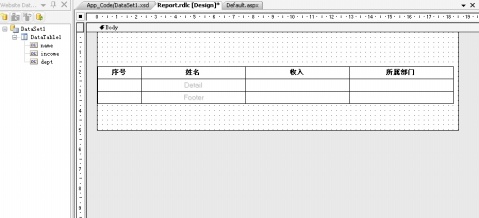
    因为报表是要数据绑定的，所以下一步添加数据源。你可以通过添加“添加数据源”来通过数据库连接字符串链接到一个表或视图，不过你也不必通过数据库来给dataset，取消弹出的“配置数据适配器”界面。打开dataset直接右键添加一个“table”（如图）

[](http://img.blog.163.com/photo/0AzfNxAtI1RnJVC21yTehA==/2294302535169039465.jpg)

然后手动加上需要的数据字段，如 name（姓名）、income（收入）、dept（部门），income的datetype为decimal。其实dataset就是一个数据外壳罢了。

三、设计报表

    从“工具栏”上拖一个“表”到报表上。此表默认有三行，header、detail、footer，顾名思义：header即是表头，detail就是要显示的内容，footer为表尾，可以在这里进行一些统计信息。并且对表格进行相应的格式设计，加上表格边宽，字体居中等，如图

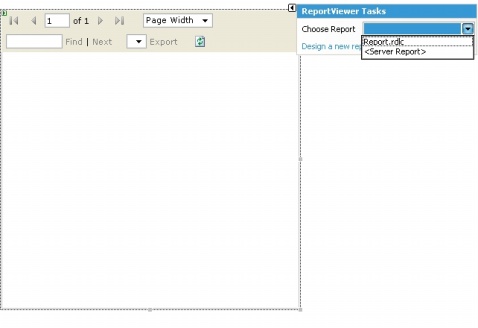
[](http://img.blog.163.com/photo/sAUFZVkAABCjHJQeAQWcgA==/2265592087544545088.jpg)

下面的工作就是把相应的数据字段帮顶上去，看到上图的左边“website data sources”了，直接把刚才建的dataset的字段拖进去，在序号列加入报表的一个函数“=RowNumber(Nothing)”，RDLC中有许多函数可以用，右击某个单元格“属性”，在“value”中点“fx”会出现函数对话框。设计成如下图：

[](http://img.blog.163.com/photo/oURcP3h45f_9TjHQTCj5kQ==/4284049145536476538.jpg)

四、设计页面

    随便建一个aspx页面，从工具栏上拖入“reportviewer”控件，点击其右上方三角，选择刚刚建立的那张报表。这是页面上会多出一个ObjectDataSource1控件是用来获取数据的，我们手动赋值，所以就不用了，删除它。

[](http://img.blog.163.com/photo/sPbGrCTXjtWp6GxLUn7cSw==/2319916758049557396.jpg)

接下来就是手动绑定数据源，具体参照以下代码：

using System;  
using System.Data;  
using System.Configuration;  
using System.Web;  
using System.Web.Security;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using System.Web.UI.WebControls.WebParts;  
using System.Web.UI.HtmlControls;  
using Microsoft.Reporting.WebForms;

public partial class \_Default : System.Web.UI.Page   
{  
    protected void Page\_Load(object sender, EventArgs e)  
    {  
        DataTable dt = GetData();

        ReportDataSource rds = new ReportDataSource("DataSet1\_DataTable1", dt);  
        ReportViewer1.LocalReport.DataSources.Clear();  
        ReportViewer1.LocalReport.DataSources.Add(rds);  
        ReportViewer1.LocalReport.Refresh();  
    }

    /// <summary>  
    ///   模拟数据  
    /// </summary>  
    /// <returns></returns>  
    private DataTable GetData() {  
        DataTable dt = new DataTable("dataname");  
        dt.Columns.Add(new DataColumn("Name", typeof(string)));  
        dt.Columns.Add(new DataColumn("Income", typeof(decimal)));  
        dt.Columns.Add(new DataColumn("Dept", typeof(string)));

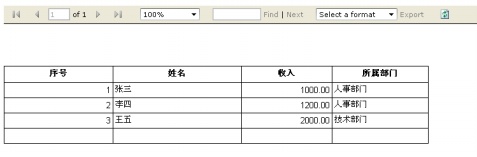
        DataRow row = dt.NewRow();  
        row["Name"] = "张三";  
        row["Income"] = 1000.00m;  
        row["Dept"] = "人事部门";  
        dt.Rows.Add(row);

        row = dt.NewRow();  
        row["Name"] = "李四";  
        row["Income"] = 1200.00m;  
        row["Dept"] = "人事部门";  
        dt.Rows.Add(row);

        row = dt.NewRow();  
        row["Name"] = "王五";  
        row["Income"] = 2000.00m;  
        row["Dept"] = "技术部门";  
        dt.Rows.Add(row);

        return dt;  
    }  
}

值得注意的是：我上面红色标出的ReportDataSource rds = new ReportDataSource("DataSet1\_DataTable1", dt);必须与html中  <rsweb:ReportDataSource DataSourceId="ObjectDataSource1" Name="DataSet1\_DataTable1" />对应，现在直接运行就可以看到效果：

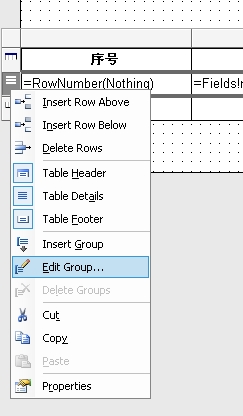
[](http://img.blog.163.com/photo/A4KDcEabBWc92owsTErobg==/5368853705779570016.jpg)

五、加统计行

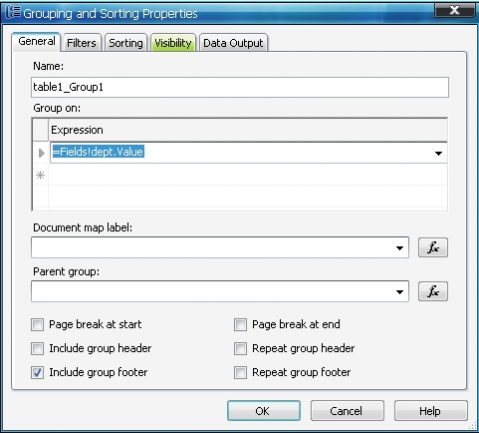
    打开报表，在footer行收入那列中键入“=Sum(Fields!income.Value)”即求和。

六、分组统计

   点击表格，在左侧上右键“inert group”，插入一个分组

[](http://img.blog.163.com/photo/KDfqIaMxPx824VWeUOz_JQ==/4287426845257004468.jpg)

在“group on”（分组依据）中选择“=Fields!dept.Value”，并且选中“include group footer”，如图：

[](http://img.blog.163.com/photo/YOjSUUONmpgsHkNZGu8rrg==/885520276732113599.jpg)

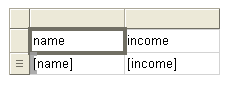
好下面再来预览一下最终效果：

[](http://img.blog.163.com/photo/5N5ikPLcUX-B2VsvVguTTg==/298363475313801707.jpg)

### 使用RDLC实用例子(分组折叠钻取跳转)

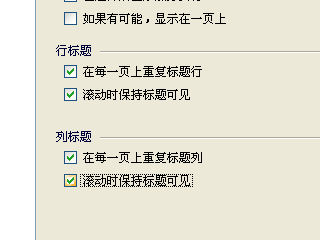
Vs2010中rdlc报表

1. ReportViewer rdlc报表分组。
2. 选择主表最左上角的方块并点击

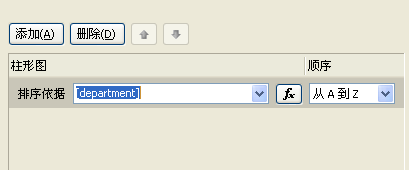


选择Tablix属性

在“常规”选项卡中选择“行标题”，“列标题”中的Checkbox.

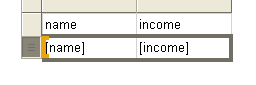


再点击“排序”选项卡，添加排序依据。

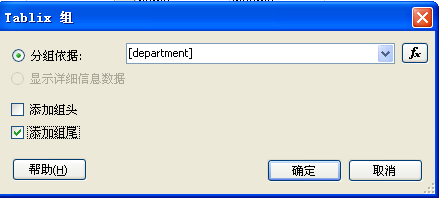


完成之后点击确定。

再点击报表中的数据行

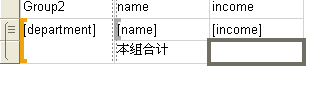


选择“添加组”行组中的“父组”

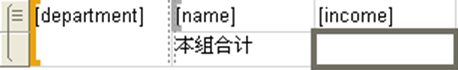


选择你的分组依据。如果要添加“组头”“组尾”可以勾选上。

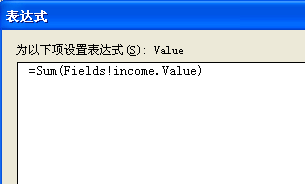
再在“组尾”上右击文本框选择“表达式”



这个就是添加了组尾的效果



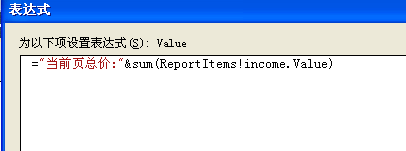
下面写入你要合计的表达式



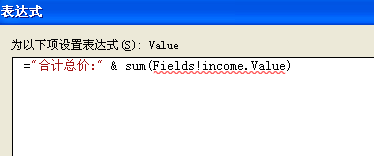
完成了。

如果想要显示每页中合计。则可以在页眉或页脚中添加一个“文本框”

在里面输入



如果想要显示总合计：



最终显示：

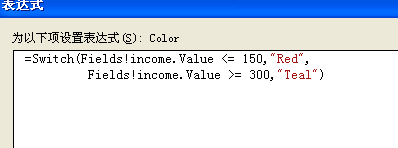


 假如要对报表中某些记录的字体设颜色，例如对总价小于100的这一行字体设红色，总价大于2000的设水鸭色。

由于是要对一行字体设颜色，所以要选中一行，点击“table1”表的详细行行头，如下图：



选择表达式设置



或：=Iif(Fields! income.Value < =150,"Red",

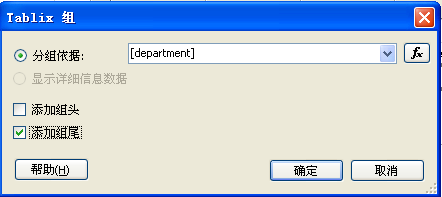
     Iif(Fields! income.Value >= 300,"Teal","Black"))

按组折叠报表

如图：

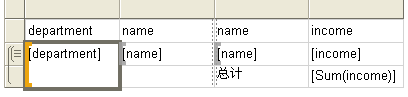


添加“父组”同上面一样。



点击确认即可。

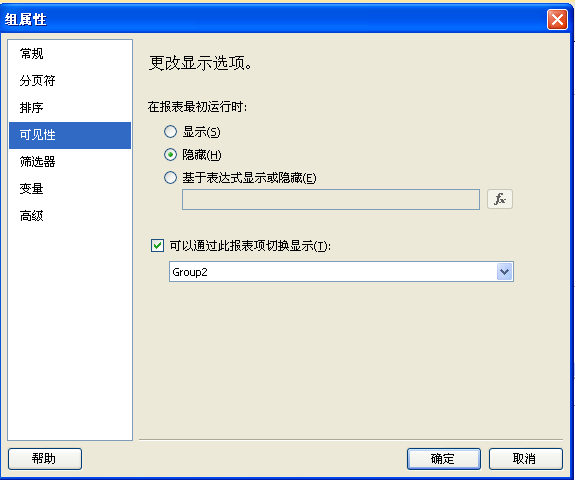
与此同时还要添加“子组”



添加子组图



在“子组”属性中选择“可见性”。做如下选择。其中Group2为文本框的名。“父组”不改变。



完成了！！

添加钻取式报表

在Page\_Load中

ReportViewer1.Drillthrough += new DrillthroughEventHandler(ReportViewer1\_Drillthrough);

/// <summary>

/// 添加钻取式报表事件

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

void ReportViewer1\_Drillthrough(object sender, DrillthroughEventArgs e)

{

LocalReport localReport = (LocalReport)e.Report;

localReport.ReportPath = MapPath("Report3.rdlc");

//这里的ReportParameter1是Report3.rdlc中参数属性的名称

//Microsoft.Reporting.WebForms.LocalReport report = e.Report as //Microsoft.Reporting.WebForms.LocalReport;

//string emp\_name = report.OriginalParametersToDrillthrough[0].Values[0].ToString();

string depName = localReport.GetParameters()["ReportParameter1"].Values[0].Trim();

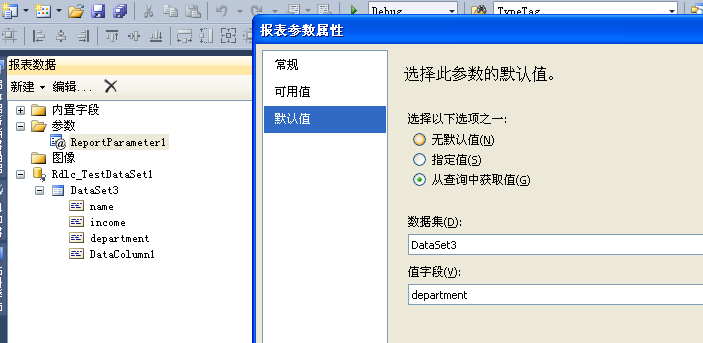
DataTable dt = new Rdlc\_TestDataSet1TableAdapters.dataNameTableAdapter().GetData(depName);

localReport.DataSources.Clear();

localReport.DataSources.Add(new ReportDataSource("DataSet3",dt));

}

在子报表中添加一个参数并设置默认值



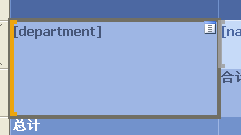
选择数据行



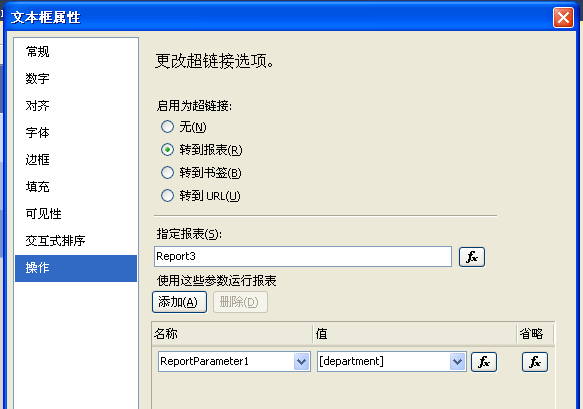
右击选择Tablix属性中的筛选器设置相关的信息



在主报表中

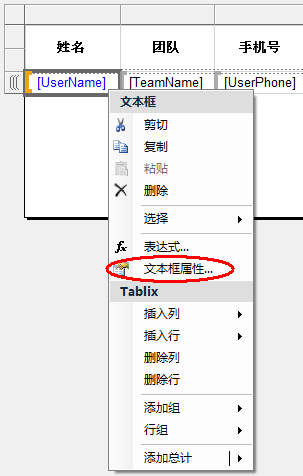
选择要文本框右击文本框属性弹出

选择操作

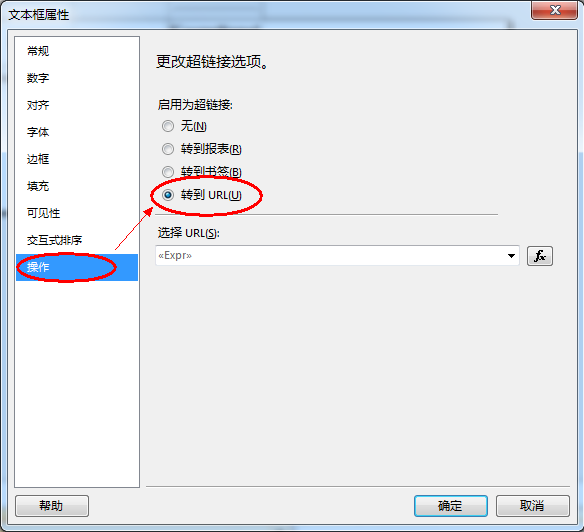


填入相对应的名称和值。

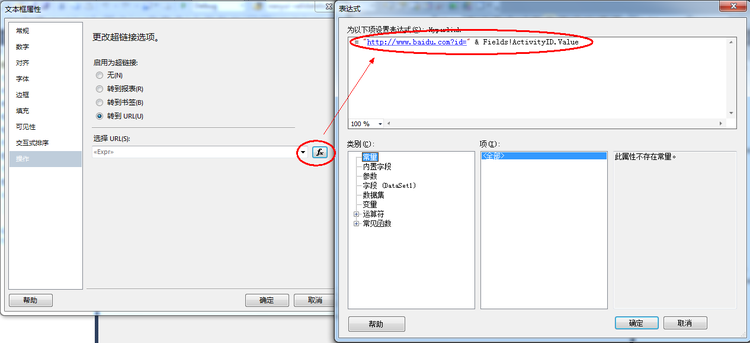
**在RDLC报表中添加链接**

第一步：选择要添加链接的文本框，点击鼠标右键，如下图：  
  
< xmlnamespace prefix ="v" ns ="urn:schemas-microsoft-com:vml" />

第二步：选择“文本框属性”，点击“操作”面板，设置右侧的“启用为超链接”选中“转到URL”，如下图：



 第三步：点击面板上的“Fx”小按钮，在弹出的面板中输入要跳转的链接地址添，若有需要可以在地址后添加参数，设置完成后，点击确定，如下图：

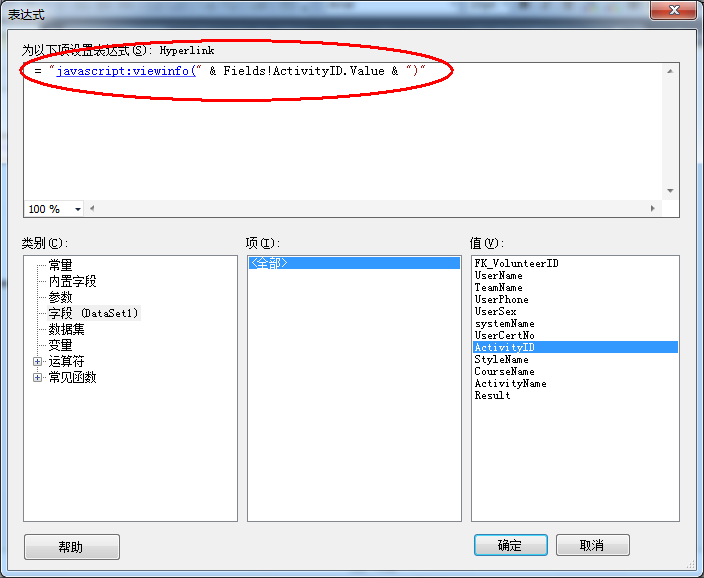


 在这里测试的输入的内容为：(要以http://开头，并出现下划线才会生效)

= ”http://www.baidu.com?id=” & Fields!ActivityID.Value

注意：地址前的“=”符号一定不能漏掉，这里的参数id的值为rdlc数据集中的已有字段ActivityID ，通过”&”符号与参数名称连接。

另外，在这里还可以不输入URL地址，而是添加一个javascript脚本，如下图：



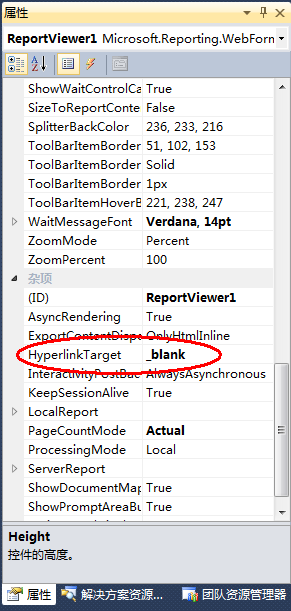
 在这里测试的输入的内容为：

= "javascript:viewinfo(" & Fields!ActivityID.Value & ")"

注意：地址前的“=”符号一定不能漏掉，这里的viewinfo是调用报表页面中已经存在的自定义函数，参数 Fields!ActivityID.Value 是数据集中的已有字段ActivityID ，通过”&”符号与参数名称连接。

第四步：返回到添加了报表控件(ReportViewer)的页面，添加如下代码：

第五步：若想点击报表中的链接，打开一个新的窗口，还要设置报表中链接的Target类型,这个设置是ReportViewer控件的一个属性，打开属性面板，找到HyperlinkTarget属性（默认值是\_top），更改它的值为“\_blank”，如下图：



 到这里，所有设置已经完成，最终效果图如下：



### RDLC常用表达式的使用

1. 连接字符

=Fields!FirstName.Value + Fields!LastName.Value 　　连接两个字段在一个单元格中，并一行显示

=Fields!FirstName.Value & vbCrLf & Fields!LastName.Value　连接两个字段在一个单元格中显示，但是换行显示

2. If表达式使用

=IIF(Fields!LineTotal.Value > 100, True, False) 如果成立执行True的表达式，否则执行False表达式，我可也可以写其它的表达式代替代码里的东西

3. Count表达式使用

=Count(Fields!DAILYCOUNT.Value,"WeldingReportDataSet\_T\_WelderDaily")

计算WeldingReportDataSet\_T\_WelderDaily数据集中DAILYCOUNT有多少行

3. Sum表达式使用

=Sum(Fields!DAILYCOUNT.Value,"WeldingReportDataSet\_T\_WelderDaily")

计算WeldingReportDataSet\_T\_WelderDaily数据集中DAILYCOUNT的总和

4. Format表达式使用

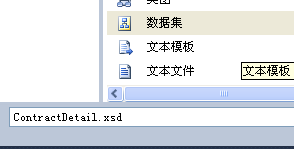
=CDate(Fields!TOPAINTDATE.Value).ToString("dd MMM yyyy") 对date的格式进行应用

六． 总结

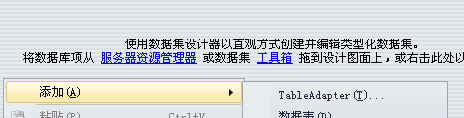
=Sum(IIF((Fields!Age.Value>20 and Fields!Age.Value<=25),1,0))

### 实例

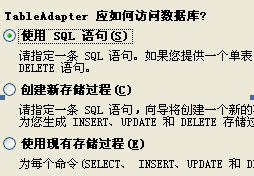
1. 创建数据集命名为ContractDetail.xsd（合同明细）



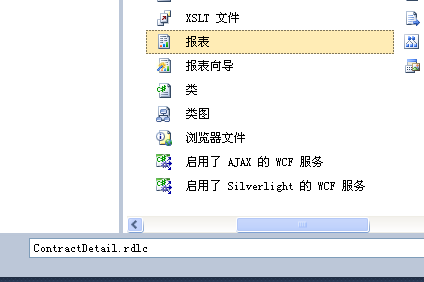
2、为数据集添加TableAdapter



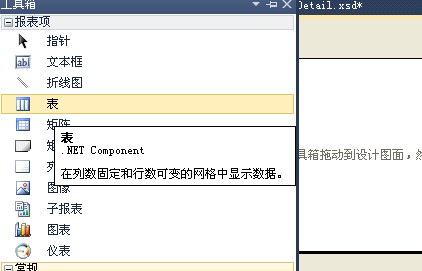
3、根据数据获取方式选择SQL或存储过程



4、然后创建报表并命名为ContractDetail.rdlc



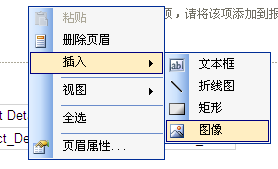
5、 拖动一张表到报表设计界面



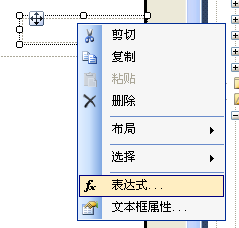
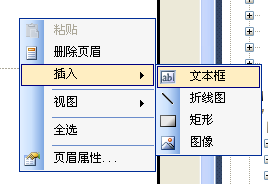
6、点击表内的表格出现添加数据源界面，并按数据集命名为ContractDetail后选择数据源



7、右击报表设计空白区域按提示， 可进行添加页眉、页脚、插入图片等操作



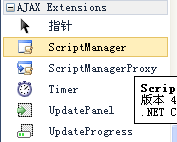
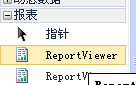
8、右击空白区域插入文本框，右击文本框-->选择fx表达式 可使用内置字段



内置字段:双击选择 ExecutionTime(列印日期) 、PageNumber(当前页数)、TotalPages

(总页数)等

9、添加字段到表里面，创建一个web页面🡪添加ReportViewer和ScriptManager用于查看报表



10、为ReportViewer报表查看控件添加 创建的报表



注意：有些情况下创建报表时无法找到数据源，可以新建一个工程单独创建数据集和报表，然后在现有工程中引入进行手动创建数据集。代码如下：

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

StringBuilder sql = new StringBuilder();

sql.Append(" SELECT (SELECT Emp\_Name FROM Emp\_Info WHERE (Emp\_Id = M.Emp\_Id)) AS Emp\_Name,");

sql.Append(" (SELECT Dept\_Name FROM Emp\_Dept WHERE (Dept\_Id = (SELECT Dept\_Id FROM Emp\_Info AS Emp\_Info\_1 WHERE (Emp\_Id = M.Emp\_Id)))) AS Dept\_Name,");

sql.Append(" [Month],OTHours,Over\_OTHours,Dept\_OTHours\_Id,");

sql.Append(" (SELECT Dept\_Hours FROM dbo.Atd\_Dept\_OTHours WHERE (Dept\_Id = (SELECT Dept\_Id FROM dbo.Emp\_Info AS Emp\_Info\_2 WHERE (Emp\_Id = M.Emp\_Id)))) AS Dept\_OTHours");

sql.Append(" FROM Atd\_OTHours\_Count AS M where [Month]=Convert(nvarchar(4),Year(SYSDATETIME()))+'-'+CONVERT(nvarchar(2),month(SYSDATETIME()))");

ReportViewer1.LocalReport.ReportPath = AppDomain.CurrentDomain.BaseDirectory + "Atd/Report/OTHours\_Count.rdlc"; //报表在工程中的路径

DataTable reqrpt = RequestReport(sql.ToString());

if (reqrpt != null)

{

ReportViewer1.LocalReport.DataSources.Clear();

ReportViewer1.LocalReport.DataSources.Add(new ReportDataSource("OTHours\_Count", reqrpt));

ReportViewer1.LocalReport.Refresh();

}

}

}

//返回查询结果赋值给报表

public DataTable RequestReport(string sql)

{

DataSet ds = SqlServerHelper.Query(sql);

if (ds != null && ds.Tables.Count > 0)

return ds.Tables[0];

else

return null;

}