# WinForms基于任务报表的帮助

任务帮助文档假设你已经熟悉.NET环境下编程，并且了解报表的基础知识以及知道一般情况下如何使用控制器。按照帮助文档中的步骤，你可以创建包含一整套**WinForms**报表功能的项目，并且对如何使用**WinForms**报表有更加深入的理解。

## 报表帮助文档

任务帮助文档假设你已经熟悉.NET环境下编程，并且了解报表的基础知识以及知道一般情况下如何使用控制器。按照帮助文档中的步骤，你可以创建包含一整套**C1Report**报表功能的项目，并且对如何使用**C1Report**报表有更加深入的理解。

注意：你需要在项目中引用以下命名空间：

* **C1.C1Report**

**CommonTasks.xml**

本章节中大部分专题都以预创建报表为例来进行说明。预创建报表保存在**CommonTasks.xml**报表模板文件中，前提是你已经安装了the Studio for WinForms samples。你可以在文档或者我的文档下的ComponentOne Samples\Studio for WinForms\ C1Report\ C1Report\XML\ CommonTasks目录中查看该文件。

### 添加图片到报表

你可以使用**C1ReportDesigner**添加非绑定或者绑定图片，并且添加图片水印。

#### 创建非绑定图片

非绑定图片是一种静态图片，例如logo图片，水印图片这类不需要保存在数据库中的图片。完成以下步骤，添加非绑定图片到你的报表中：

1. 打开**C1ReportDesigner。**

2. 创建一个新报表或者打开已有报表，当你在**C1ReportDesigner**中打开报表之后，你就可以对其进行修改。

3. 单击**Design**按钮，开始编辑报表。

4. 在设计模式下，单击**Add Unbound Picture**按钮，该按钮位于**Design**选项卡中的**Fields**分组中。

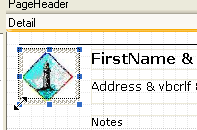


弹出**Open**对话框。

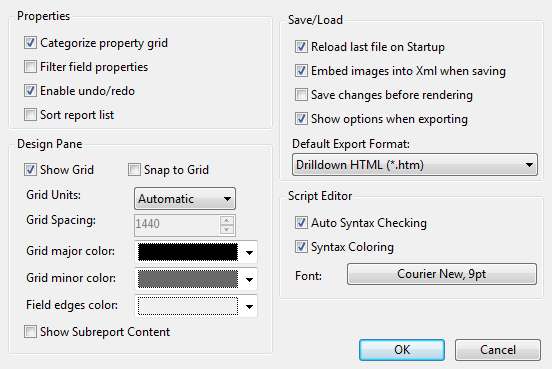
5. 选择想要在报表中包含的图片文件，并单击**OK**按钮。

6. 在报表中单击选择你希望添加图片的位置，然后调整图片框大小来显示图片。

下图显示图片已经添加到报表中，正在调整图片大小的效果：



需要注意的是，图片文件可以被嵌入到报表模板中，也可以作为一个外部文件被引用。选择你需要的方式，在**Designer**中选择**Application**按钮，在出现的菜单中选择**Options**选项。**C1ReportDesigner Options**对话框将会弹出，你可以在该对话框中选择**Embed images into Xml when saving**选项：



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"03: Unbound Images"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 创建绑定图片

绑定图片是一种保存在数据库字段中的图片类型。想要在报表中显示这些图片，你需要在报表中新增一个字段，然后设置字段的**Picture**属性，将图片存储路径添加到名称列中。

##### 使用C1ReportDesigner添加绑定图片到报表中：

1. 在**C1ReportDesigner**的设计模式下，单击**Add Bound Picture**按钮，该按钮位于**Design**选项卡中的**Fields**分组。



这里将显示一个菜单，其中包括数据源中所有的二进制字段。

1. 选择你想要加入报表中的字段。

##### 使用代码添加绑定图片到报表中：

如果数据库中的“**Photo**”字段包含内嵌的OLE对象或者原始图像流，并且报表中存在“**fEmployeePhoto**”字段。这种情况下，使用下述代码将会在字段中显示内嵌的照片。

* **Visual Basic**

fEmployeePhoto.Picture = "Photo"

* **C#**

fEmployeePhoto.Picture = "Photo";



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"04:Bound Images"章节，该文件

保存在**ComponentOne Samples**文件目录下。

#### 创建水印

水印是一种显示在报表内容下的图片类型。这种图像通常会变浅，以防止它们干扰实际的报表内容。

想要显示水印图片你需要设置图片包含文件中**Picture**属性。你还可以使用 **PictureAlign** 和**PictureShow**属性控制水印的缩放比例和它是否应该出现。



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"05: Watermark"章节，该文件

保存在**ComponentOne Samples**文件目录下。

### 创建报表字段

当报表模板加载到组件中并且完成数据源定义之后，你可以新增和编辑报表字段。你可以很简单的使用C1ReportDesigner下的Design选项卡中的Fields分组来实现该功能。



想要在报表中新增一个字段，你需要完成以下步骤：

在报表中拖拽鼠标直到光标变成十字。单击然后通过拖拽定义新字段所占的矩形位置。然后松开按钮创建新字段。



如果你改变主意，单击**ESC**键或者单击**Undo**按钮取消操作。

注意： **C1report**仅有一种**Field**对象。你可以通过设置**Field**对象的属性来使其在某一指定方式下显示和工作。

你可以通过复制、粘贴已有的字段来创建新字段。或者按住**CTRL**键，然后拖拽一个字段或者字段组到新位置来创建一个副本。

#### 创建图表

在最初版本的**C1Report**中，在报表中添加图表需要使用**StartSection**事件生成图表，然后将图表图片在字段的**Picture**属性中进行设置。这样并不难，而且还是动态添加图片到报表中最灵活的方式。

然而，这种方式有两个缺点：

* 需要在报表模板外部编写代码实现，这意味着只有你的应用才能够按照预计的方式显示报表。
* 需要编写代码生成报表，这一过程非常冗长乏味。

当前版本的**C1Report**支持自定义报表字段，包括基于**C1Chart**控制器的图表字段。想要添加一个图表区域到报表分组的页眉区域，你需要完成以下步骤：

1. 打开**C1ReportDesigner。**
2. 创建一个新报表或者打开已有报表，当你在**C1ReportDesigner**中打开报表之后，你就可以对其进行修改。
3. 单击**Design**按钮，开始编辑报表。
4. 在**Design**选项卡中的**Fields**分组中，单击**Add Chart Field**按钮。



1. 单击报表的分组页眉区域，拖动区域使其适应报表。
2. 在属性窗口中，将图表区域的**ChartDataX**和**ChartDataY**属性值设置为你想在报表中展示的数值。你可以通过将**ChartDataY**属性设置为一个通过分号间隔的字段列表（例如“UnitsInStock;ReorderLevel"），从而显示一系列的数值。

图表数据会自动根据当前报表分组进行调整。例如，当呈现“**Beverages**”区域时，只有该分类中的数据会参与绘图。你可以使用很多其他属性来定制图表，例如**Chart.ChartType**，**Chart.GridLines**，**Chart.Use3D**以及**Chart.Palette**等属性。

更多关于创建图表字段的信息，请参阅**Adding ChartFields**章节。



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"11: Charts"章节，该文件

保存在**ComponentOne Samples**文件目录下。

#### 创建自定义字段

你可以创建自定义字段，并将他们添加到**ReportDesigner**的调色板中。完成以下步骤，实现该过程：

1. 创建一个自定义字段类，该类来自C1.Win.C1Report.Field。

2. 在**Report Designer**的设置文件中登记你的自定义字段。

下面将介绍图表和渐变字段是如何实现的。这些自定义字段的源代码是可用的，你可以在创建自定义字段时使用它们作为起始点。图表和渐变字段在**C1ReportDesigner**的设置文件中使用如下代码进行注册：

<customfields>

<item value="C1.Win.C1Report.CustomFields;C1.Win.C1Report.CustomFields.Chart"/>

<item value="C1.Win.C1Report.CustomFields;C1.Win.C1Report.CustomFields.Gradient"

/>

</customfields>

例如，想要在Design调色板中增加一个新字段，需要在"C1ReportDesigner.2.exe.settings" 文件中的<**customfields**>区域增加你的控制器：

<customfields>

<item

value="C1.Win.C1Report.CustomFields.2;C1.Win.C1Report.CustomFields.Chart"

/>

<item

value="C1.Win.C1Report.CustomFields.2;C1.Win.C1Report.CustomFields.Gradien t" />

<!-- THISLINEWILLADDA NEW FIELDTO THE DESIGNER-->

<itemvalue="MyCustomFieldAssembly;MyCustomFieldAssembly.MyField"/>

</customfields>

这里假设你的字段叫做“**MyField**”，你可以在“**MyCustomFieldAssembly**”集合中找到它。

**示例报表**



完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"12: Custom Fields"章节，该文件保存在ComponentOne Samples文件目录下。

### 定制页面标题

本部分将介绍如何定制页面标题的行为。

#### 在页面分页处标题中增加连续标签

如果页眉的**Repeat**属性设置为**True**，分组页眉在页面分页时将会不断重复。这使报表更容易阅读，但就变得很难使用标题分辨出这是一个新分组页面或者是一个连续页面。

一种解决方式是增加一个名为“**Continued**”标签字段，**fContinued**，例如添加到分组页眉，通过脚本控制它的可见性。完成以下步骤，实现该功能：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**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**Detail**选项。
5. 找到**Detail.OnPrint**属性，单击旁边空白字段，然后单击**ellipsis**按钮。
6. 在弹出的**VBScript**编辑器中，输入下述VB表达式：
7. 在属性窗口上方的下拉列表中选择**GroupFooter**选项。
8. 找到**GroupFooter.OnPrint**属性，单击旁边空白字段，然后单击**ellipsis**按钮。
9. 在弹出的**VBScript**编辑器中，输入下述VB表达式：

' VBScript:GroupFooter.OnPrint

fContinued.Visible= false

如果**fContinued**字段初始化为不可见，脚本只会在连续页面的页眉中显示该标签。脚本确认**fContinued**字段在分组中是可见的。任何在分组页脚之后，下一个详细区域之前创建的页面分割，都不会显示该标签。



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"18: Continued Headers"章节，该文件保存在ComponentOne Samples文件目录下。

#### 动态改变页面页眉

你可以使用**C1Report**的**PageHeader** 和 **PageFooter**属性来指定页面页眉还是页面页脚区域是否显示在所有页面，或者限制页面是否包含在报表页眉和页脚区域。

某些情况下，你想要进一步定制这种行为。例如，你想要在奇数页面和偶数页面呈现不同的页眉。可以使用脚本根据页面被呈现的情况选择显示或者隐藏字段。完成以下步骤实现该功能：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**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**Detail**选项。
5. 找到**OnFormat**属性，单击旁边空白字段，然后单击**ellipsis**按钮。
6. 在弹出的**VBScript**编辑器中，输入下述VB表达式：

odd= (pagemod2<> 0) h1odd.Visible= odd

h2odd.Visible= odd

h1even.Visible= not odd h2even.Visible= not odd

如果报表页眉包含“h<x>odd”和“h<x>even”字段，该脚本将根据页面是奇数还是偶数选择显示或者隐藏字段。

需要注意的是，为了安置页面页眉显示空白区域，所有的字段的**CanShrink**属性都需要设置为**True**。



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"09: Dynamic Page Header"章节，该文件保存在**ComponentOne Samples**文件目录下。

### 自定义页面布局

下面将介绍如何定制报表的布局。

#### 控制页面分页

缺省情况下，**C1Report**将在每个页面底部插入一个分页符，然后继续呈现下一个页面。你使用下述几个属性可以重载这一行为：

* Group.KeepTogether：该属性决定分组页眉是否在页面中呈现，如果它们必须在至少一个Detail区域呈现，或是全部的分组应该显示在同一个页面。
* Section.KeepTogether：决定是否允许页面分页。该属性优先级低于Group.KeepTogether。
* ForcePageBreak：允许你指定页面分页应该插入在分组之前或者之后，还是区域之前或者之后。
* Field.KeepTogether：决定字段是否允许页面分页。该属性允许长文本字段横跨多个页面。它的优先级低于Section.KeepTogether。
* ForcePageBreak：允许你指定页面分页应该插入在分组之前或者之后，还是字段之前或者之后。

你可以通过**C1ReprotDesigner**的属性表格设置这些属性。

你可以在报表呈现过程中使用脚本改变这些属性。例如，每个10个**Detail**区域执行一次页面分页，完成以下步骤实现该功能：

1. 打开**C1ReportDesigner**，想要了解**C1ReprotDesigner**更多信息，请参阅[**Accessing C1ReportDesignerfrom VisualStudio**](#_bookmark102)**。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**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**Detail**选项。
5. 找到**Detail.OnPrint**属性，单击旁边空白字段，然后单击**省略号**按钮。
6. 在弹出的**VBScript**编辑器中，输入下述VB表达式：

cnt= cnt+1 detail.forcepagebreak="none"

if cnt>= 10 then

cnt=0

detail.forcepagebreak="after"

endif



**示例报表**

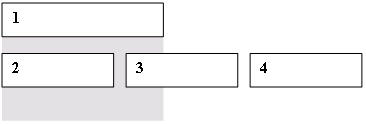
完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"07: Force Page Breaks"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 创建CanGrow/CanShrink字段

报表字段的内容横跨多行或是一行都不占是很常见的事情。某些情况下，你也许希望允许字段扩充或者缩短来适应其内容，总好过在报表中切割多出的长度或是剩下白色空白。

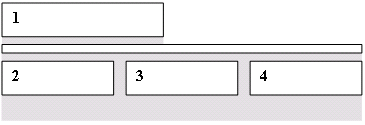
想要完成该功能，在C1ReportDesigner的设计模式下，设置Field对象的CanGrow 和CanShrink属性为True。

字段区域扩充时可以将处于下方的字段内容下推。同样的，收缩时也可以将处于下方的字段内容上移。这里的下方是严格意义上的下方，就像下面对话框中显示的那样：



随着字段1的扩充或是收缩，字段2,3将进行上移或是下移。字段4不受影响，因为它并不处于字段1的下方。上图中的阴影区域显示的是受字段1影响的区域。

如果你想要字段4和字段2,3保持一致，你需要添加一个格外的字段横跨字段2和字段3的上方。字段1将推动新字段，从而推动字段2,3,4。下面的对话框显示这种新布局：



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"06: CanGrow CanShrink"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 创建装订线边距

装订线边距是一个额外区域，主要添加在装订线边缘位置。它使页面装订成文件或是小册子等操作变得简单。

想要在报表中增加一个装订线边距，你需要提高奇数页**MarginLeft**的属性值，然后在偶数页使用缺省数值。你可以通过脚本实现该操作。想要添加一个脚本改变呈现中页面的装订线边距，你需要完成以下步骤：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**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**Detail**选项。
5. 找到**Detail.OnPrint**属性，单击旁边空白字段，然后单击**ellipsis**按钮。
6. 在弹出的**VBScript**编辑器中，输入下述VB表达式：

' VBScript:Report.OnOpen

gutter= report.layout.marginleft ' initializevariable

' VBScript:Report.OnPage

report.layout.marginleft=\_

Iif(pagemod2= 1, gutter,gutter - 1440)



**示例报表**

完整的报表，可参阅在报表模板文件下**CommonTasks.xml**报表模板文件的"10: Gutter"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 定义和使用全局变量

报表中定义和使用全局变量并没有特殊的方式，但是你可以添加隐藏字段到报表中，然后使用它们的值作为全局参数。完成以下步骤，实现该功能：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**按钮，开始编辑报表。
4. 在**Design**选项卡中的**Fields**分组中，单击**Add Label**按钮，增加一个字段到报表中。
5. 在报表中，单击你想要放置字段的位置，然后拖拽使其适应字段内容。
6. 为字段选择以下属性：

Field.Name = **linesPerPage**

Field.Text = **14**

Field.Visible = **False**

1. 可以使用脚本来控制每个页面中Detail行的数量。在属性窗口上方的下拉列表中选择**Detail**选项。
2. 找到**OnPrint**属性，然后单击旁边空白字段，单击**ellipsis**按钮。VB脚本编辑器将会弹出。
3. 在脚本代码编辑器中输入下面的VB表达式：

cnt= cnt+1 detail.forcepagebreak="none" if cnt>=**linesPerPage**then

cnt=0

detail.forcepagebreak="after"

Endif

需要注意的是，通过修改字段的**Text**属性，你可以在报表呈现前设置**linesPerPage**字段的值。



**示例报表**

完整的报表，可参阅在报表模板文件下CommonTasks.xml报表模板文件的"08: Global Constant"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 指定自定义页面大小

缺省情况下，**C1Report**使用默认打印机中默认页面大小创建报表。

你可以使用**PaperSize**和**Orientation**属性来指定页面大小和方向。然而，**C1Report**在呈现前将会检查选中页面大小是否符合当前打印机规定。如果选中页面无效，则将页面参数改为缺省参数。

如果你想要指定一个具体页面的大小，不管是否符合打印机规定，你可以将**PaperSize**属性设置为**Custom**，然后将**CustomWidth**和**CustomHeight**属性设置为实际页面的规模（以像素为单位）。

##### 使用C1ReportDesigner指定定制页面大小为25x11：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**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**Detail**选项。
5. 找到**Layout**选项，然后张开属性节点访问所有的可用属性。
6. 将**CustomHeight**属性设置为25或是25in。

注意，尺寸将自动转换成像素单位。属性窗口将尺寸显示为36000（以像素为单位）。

1. 将**CustomWidth**属性设置为11或是11in

属性窗口将尺寸显示为15840（以像素为单位）。

1. 将**PaperSize**属性设置为**Custom**。

##### 使用代码指定定制页面大小为25x11：

不管打印机页面大小的设置，下面的代码会将报表页面大小设置为25x11：

* **Visual Basic**

c1r.Layout.PaperSize= PaperKind.Custom c1r.Layout.CustomHeight= 25\* 1440 ' in twips c1r.Layout.CustomWidth= 11\* 1440

* **C#**

c1r.Layout.PaperSize = PaperKind.Custom; c1r.Layout.CustomHeight = 25 \* 1440; // in twips c1r.Layout.CustomWidth = 11 \* 1440;



**示例报表**

完整的报表，可参阅在报表模板文件下CommonTasks.xml报表模板文件的"02: Custom Paper Size"章节，该文件保存在**ComponentOne Samples**文件目录下。

### 格式化报表

下面的内容主要介绍在报表中如何应用格式化。很简单，你可以修改属性窗口中的属性或是在VB脚本表达式中增加几行脚本就能够修改你的报表。

#### 增加交替变换的背景色

在报表中增加交替变换的背景色，你可以使用**Detail**区域的**OnPrint**属性来改变这一区域的**BackColor**属性实现这一功能。完成以下步骤实现该功能：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**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**Detail**选项。
5. 找到**OnOpen**属性，输入cnt = 0。这里是初始化cnt变量。
6. 下一步，在属性窗口上方的下拉列表中选择**Detail**选项。
7. 找到**OnPrint**属性，单击旁边的空白字段，然后单击**ellipsis**按钮。
8. 弹出VB脚本编辑器，然后在代码编辑框中输入下方的VB脚本表达式：

cnt= cnt+1

if cnt mod2=0 then

detail.backcolor= rgb(200,220,200)

else

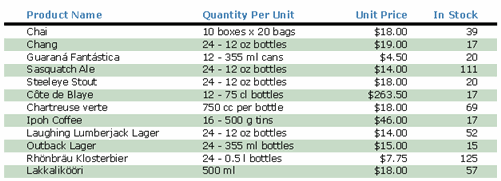
detail.backcolor= rgb(255,255,255)

endif

9. 单击**Preview**按钮，预览交替变换背景色的报表效果。

实现效果如下图所示：

下面的报表实现了交替变换的背景色。



当**Detail**区域被呈现时，计数器随之增加，**BackColor**属性将会自动切换。



**示例报表**

完整的报表，可参阅在报表模板文件下CommonTasks.xml报表模板文件的"01: Alternating Background (Greenbar report)"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 增加条件格式化

在某些情况下，你也许想要根据字段内容改变一个字段的显示效果。例如，你也许想要让仓库中贵重的物品或是便宜的物品高亮显示。下面的脚本可以实现这一功能。

完成以下步骤，实现该功能：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**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**Detail**选项（因为该区域包含想要添加格式化条件的字段）。
5. 找到**OnFormat**属性，单击旁边的空白字段，然后单击**ellipsis**按钮。
6. VB脚本编辑框将会弹出，在代码编辑框中输入下述VB脚本表达式：

' VBScript: Detail.OnFormat

If UnitsInStock + UnitsOnOrder < ReorderLevel And \_

Discontinued = False Then Detail.BackColor = rgb(255,190,190)

Else

Detail.BackColor = vbWhite

Endif

脚本将会根据**UnitsInStock, UnitsOnOrder, ReorderLevel和Discontinued**的值改变**Detail**区域**BackColor**属性。



**示例报表**

完整的报表，可参阅在报表模板文件下CommonTasks.xml报表模板文件的"16: Conditional Formatting"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 根据字段值编辑字段格式

你可以通过为**Detail**区域的**OnFormat**属性指定一个表达式来实现根据字段值改变报表中字段格式的功能。

想要为**OnFormat**属性指定表达式，你需要完成以下步骤：

1. 打开**C1ReportDesigner。**
2. 创建一个报表或者打开已有报表，在**C1ReprotDesigner**中打开报表之后，你就可以修改它了。
3. 单击**Close Print Preview**按钮，开始编辑报表。
4. 在设计模式下，从属性窗口的下拉列表中选择**Detail**选项，从而查看**Detail**区域所有可用属性。
5. 找到**OnFormat**属性，然后单击属性旁边的**ellipsis**按钮。
6. VB脚本编辑框将会弹出，你可以在此指定一个表达式。

如果**UnitsInStock**和**UnitsOnOrder**数值总和小于**ReorderLevel**的值，下面的表达式将改变**UnitsInStock**字段的前景色为红色。下面有几种书写表达式的方式：

**选择1:**

UnitsInStockCtl.Forecolor = Iif(UnitsInStock + UnitsOnOrder < ReorderLevel, vbRed, vbBlack)

**选择2:**

lowStock = UnitsInStock + UnitsOnOrder < ReorderLevel UnitsInStockCtl.Forecolor = Iif(lowStock, vbRed, vbBlack)

**选择3:**

If UnitsInStock + UnitsOnOrder < ReorderLevel Then UnitsInStockCtl.Forecolor = vbRed

Else

UnitsInStockCtl.Forecolor = vbBlack

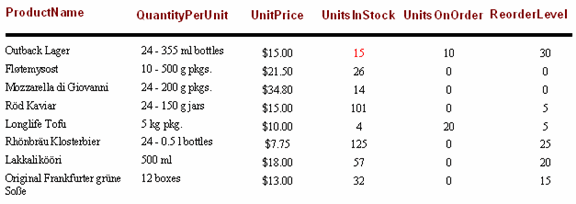
End If

**选择4:**

color = Iif(UnitsInStock + UnitsOnOrder < ReorderLevel, vbred, vbblack) UnitsInStockCtl.Forecolor = color

实现效果如下所示：

注意，因为**UnitsInStock**和**UnitsOnOrder**的总和小于**ReorderLevel**，**Outback Lager**的**UnitsInStock**数值变为红色：



#### 禁止或强制全零数组的显示

你可以将字段的**Format**属性设置为#，从而禁止显示数值为0的字段。#号是一个格式标识，用于指定仅显示有效数值（前面或者后面不加零）。

使用例如“0000”这样的格式，来强制显示某一具体数值。这里的零将强制显示数据，无论前面还是后面添加零。

每一种格式字符串都通过分号分割成三部分。如果尝试使用两部分，第一部分用于正值和0，第二部分用于负值。如果尝试使用三部分，第一部分用于正值，第二部分用于负值，第三部分用于0。例如，“#;(#);ZERO”。



**示例报表**

完整的报表，可参阅在报表模板文件下CommonTasks.xml报表模板文件的"21: Suppress or Force Zeros"章节，该文件保存在**ComponentOne Samples**文件目录下。

### 加载报表模板

**C1Report**使用行数据组合报表模板来完成创建报表的工作。为了创建报表，你需要加载报表模板到**C1Report**中。下面将介绍几种加载报表模板的方式。

#### 从文件中加载报表模板

你可以使用**C1ReportDesigner**来加载报表模板文件（XML文件中将包含一个或者多个报表模板）。

##### 在设计阶段从文件中加载报表模板：

在设计阶段从文件中加载报表模板，你需要完成下述任务中的一种：

* 右键单击**C1Report**组件，选择**Load Report**菜单选项

或者

* 单击**C1Report**组件上的**smart**标签，然后从**C1Report**任务菜单中选择**Load Report**选项。

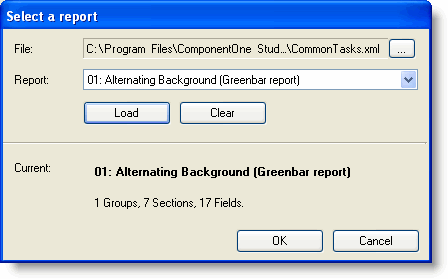
在**Select a report**对话框中选择你想要加载的报表，完成以下任务：

1. 单击**ellipsis**按钮，**Open**对话框弹出，你可以选择XML文件。

2. 可用报表模板将在**Report**的下拉列表中显示。选择一个报表模板进行加载。

3. 单击**Load**按钮，然后单击**OK**按钮关闭对话框。

报表选择对话框效果如下所示：



##### 使用代码从文件中加载报表模板

你可以使用**Load**方法，从文件中加载报表模板。将你希望加载报表的报表模板文件名称和报表名称作为参数。如果你想要罗列报表模板中的报表，可以使用**GetReportInfo**方法。它将以列表的形式返回文件中所有报表。

* + **Visual Basic**

' Get list of reports in a report definition file

Dim reports As String() = c1r.GetReportInfo(reportFile)

' Load first report into C1Report component

c1r.Load(reportFile, reports(0))

* + **C#**

// Get list of reports in a report definition file

string[] reports = c1r.GetReportInfo(reportFile);

// Load first report into C1Report component

c1r.Load(reportFile, reports[0]);

#### 通过字符串加载报表模板：

**C1Report**有一个**ReportDefinition**属性，该属性允许你以字符串的形式获取或者设置完整报表模板。这是一种非常方便的实现方式，你可以在数据库或者自身应用的数据结构中对报表模板进行存储和检索。

**ReportDefinition**字符串中包含了一个完整的XML，并且存储在报表模板文件中。举例说明如下：

* **Visual Basic**

' Load report definition into C1Report component

c1r.Load(reportFile, reportName)

' Copy report definition to the clipboard

Dim repDef As String = c1r.ReportDefinition Clipboard.SetDataObject(repDef)

' Copy report definition to c1r2 component

c1r2.ReportDefinition = repDef

* **C#**

// Load report definition into C1Report component

c1r.Load(reportFile, reportName);

// Copy report definition to the clipboard

string repDef = c1r.ReportDefinition; Clipboard.SetDataObject(repDef);

// Copy report definition to c1r2 component

c1r2.ReportDefinition = repDef;

### 修改子报表

本部分主要介绍如何修改子报表。子报表是指正式报表中包含了另外一个报表的字段。主报表经常设计用于作为显示主报表中详细信息的主要场景。

#### 子报表中增加页眉

**C1Report**在子报表中不显示页眉和页脚。它仅在主报表中使用。在**Microsoft Access**中同样如此。

在多数情况下，你可能想要在子报表中页面分页中包含页眉信息。想要实现该功能，你需要将页眉放入分组页面区域，并且设置区域的**Repeat**属性为**True**。如果你的子报表并不包含分组，请添加一个空白分组。



**示例报表**

完整的报表，可参阅在报表模板文件下CommonTasks.xml报表模板文件的"14: Page Headers in Subreports"章节，该文件保存在**ComponentOne Samples**文件目录下。

#### 在子报表中检索数据

在某些情况下，你想要将子报表中的数据传递给主报表。脚本变量并不能实现该功能，因为每个报表都有自己的脚本作用域（这将避免变量名冲突的可能）。

想要将子报表中的数据传递给主报表，你必须将值存储在子报表字段中或者在子报表的Tag属性中，然后让主报表阅读这些值。

在这个例子中，子报表计算出了每个商品类别的平均价格，然后将其存储在Tag属性中。主报表检索并显示这个值。



**示例报表**

完整的报表，可参阅在报表模板文件下CommonTasks.xml报表模板文件的"15: Retrieve Values from Subreports"章节，该文件保存在**ComponentOne Samples**文件目录下。

### 呈现报表（预览，打印以及导出）

一旦报表模板加载到组件中，并且完成数据源的定义，你就可以将报表呈现到打印机，预览控制器或者是导出成报表文件。

#### 在报表呈现过程中显示进度显示器

大多数预览应用在呈现页面时都包含进度提示，并且拥有一个按钮允许你取消报表生成。.NET打印预览控制器自动提供这一功能。如果你直接打印报表或者导出报表文件，这里并没有内置的进度显示在报表UI中。

你可以使用**C1Report**事件创建一个报表进度对话框，或者在呈现报表时更新状态条。起始页和结束页事件将会有足够的反馈告诉你哪个页面正在被打印以及报表什么时候完成。例如，下述代码使用起始页事件给状态条(StatusStrip1)提供反馈。

* **Visual Basic**

Private Sub c1r\_StartPage(ByVal sender As System.Object, ByVal e As C1.Win.C1Report.ReportEventArgs) Handles c1r.StartPage

StatusStrip1.Text = String.Format("Rendering page {0} of '{1}'...", c1r.Page, c1r.ReportName)

End Sub

* **C#**

private void c1r\_StartPage(object sender, ReportEventArgs e)

{

statusStrip1.Text = string.Format("Rendering page {0} of '{1}'...", c1r.Page, c1r.ReportName);

}

想要在报表完成之前取消动作，增加一个取消按钮到你的应用中，然后使用它来设置**C1Report**的**Cancel**属性为**True**。例如：

* **Visual Basic**

Private Sub \_btnCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click

c1r.Cancel = True

Close()

End Sub

* **C#**

private void \_btnCancel\_Click(object sender, System.EventArgs e)

{

c1r.Cancel = true;

Close();

}

需要注意的是你也许还想要提供进度条以及当前是第几页的提示，但是这通常情况下很难去实现。因为页面计数器在报表呈现完成前并不知道当前正在呈现的页数。



**示例报表**

完整的报表进度指示例子，可参阅 **Progress Indicator**例子，该文件保存在**ComponentOne Samples**文件目录下。

#### 预览报表

想要预览报表，可以使用**C1Report.Document**属性。在**Report for WinForms**预览控制器中或者在.NET打印预览或者打印预览对话框控制器中设置**Document**属性。预览控制器将显示该报表，并且允许用户区浏览，缩放或者打印报表。例如：

* **Visual Basic**

' Load report definition

c1r.Load(reportFile, reportName)

' Preview the document

c1preview1.Document = c1r.Document

* **C#**

// Load report definition

c1r.Load(reportFile, reportName);

// Preview the document

c1preview1.Document = c1r.Document;

**注意：C1Report**使用.NET预览组件实现该功能，但是它充分利用了内置的**Reports for WinForms**预览控制器。当你使用内置控制器时，你可以在页面生成时看到该报表页面。当你使用标准控制器时，你必须等到全部报表完成渲染之后才能查看第一个页面。

#### 打印报表

你可以使用C1Report.Document属性来直接打印报表。这一属性返回一个标准PrintDocument对象，该对象拥有一个Print方法以及公开的打印机和页面设置。

例如，下述代码显示一个打印对话框，然后打印报表：

* **Visual Basic**

' Load report definition

c1r.Load(reportFile, reportName)

' Get PrintDocument object

PrintDocument doc = c1r.Document

' Show a PrintDialog so user can customize the printing

Dim pd As PrintDialog = New PrintDialog()

' Use PrinterSettings in report document

pd.PrinterSettings = doc.PrinterSettings

' Show the dialog and print the report

If pd.ShowDialog() = DialogResult.OK Then doc.Print()

End If

' Cleanup and release PrintDialog resources

pd.Dispose()

* **C#**

// Load report definition

c1r.Load(reportFile, reportName);

// Get PrintDocument object

PrintDocument doc = c1r.Document;

// Show a PrintDialog so user can customize the printing

PrintDialog pd = new PrintDialog();

// Use PrinterSettings in report document

pd.PrinterSettings = doc.PrinterSettings;

// Show the dialog and print the report

if (pd.ShowDialog() == DialogResult.OK) doc.Print();

// Cleanup and release PrintDialog resources

pd.Dispose();

#### 导出报表

##### 按正常文件格式导出报表

**C1Report**有一个**RenderToFile**方法，允许你按照几种不同的文件格式导出报表，其中包含HTML, RTF, PDF, TIFF, Text, 和XLS等等。例如，下述代码创建一个PDF和XLS版本的报表：

* + **Visual Basic**

' Load report definition

c1r.Load(reportFile, reportName)

' Export to PDF

c1r.RenderToFile(outFile + ".pdf", FileFormatEnum.PDF) c1r.RenderToFile(outFile + ".xls", FileFormatEnum.Excel)

* + **C#**

// Load report definition

c1r.Load(reportFile, reportName);

// Export to PDF

c1r.RenderToFile(outFile + ".pdf", FileFormatEnum.PDF);

c1r.RenderToFile(outFile + ".xls", FileFormatEnum.Excel);

**注意：**如果在“**preserve pagination**”选项选中的情况下，一个文件导出成RTF或者DOCX格式的文件，文本将被置于文本框中并且从结果文档中恢复文本的功能将被限制。

##### 按自定义格式导出报表

如果**C1Report**并不支持你想要导出的报表格式，你可以自己编写导出过滤器类，使用**C1Report**.**RenderToFilter**方法将报表呈现到自定义的过滤器中。

自定义过滤器类继承于**C1.Win.C1Report.ExportFilter**类。通过重载少部分简单方法，例如：**StartReport**, **StartSection**, **RenderField**, **EndSection,** 和**EndReport**来实现该功能。

写一个自定义导出过滤器并不困难。例如，它可以用于创建自定义XML格式的报表，以用于其他应用的后续处理。

### 保存报表模板

当你在**C1ReportDesigner**应用中完成创建和浏览报表，选择**Application**按钮然后选择**Save**选项能够保存你的报表模板文件。**Designer**将按照XML格式保存报表模板，该格式可以在**Designer**中阅读或者直接导入到**C1Report**组件中。