### 插入分页符

通过RenderObject的BreakAfter插入一个分页符。

1. 从工具栏上向您的工程添加C1PrintPreviewControl 和C1PrintDocument控件。
2. 单击C1PrintPreviewControl1 选中至属性窗体，设置其Document属性为

###### C1PrintDocument1。

1. 添加以下代码至Form\_Load事件：
	* VisualBasic

'构建文档。

MakeDoc()

'生成文档。

Me.C1PrintDocument1.Generate()

* + C#

//构建文档。

MakeDoc();

//生成文档。

this.c1PrintDocument1.Generate();

1. 添加 MakeDoc 子方法，并通过BreakAfter属性在每一个RenderObject之后插入一个分页符：
	* VisualBasic

PrivateSub MakeDoc()

'创建 RenderText。

Dim rt1 As New C1.C1Preview.RenderText

rt1.Text= "Thisis RenderText.A RenderImagewillbe on page2and

a RenderGraphicon page3."

'添加一个分页符。

rt1.BreakAfter= C1.C1Preview.BreakEnum.Page

'创建RenderImage。

Dim ri1 As New C1.C1Preview.RenderImage

ri1.Image= System.Drawing.Image.FromFile("c:\c1logo.bmp")

'添加一个分页符。

ri1.BreakAfter= C1.C1Preview.BreakEnum.Page

'创建一个RenderGraphic。

Dim rg1 As New C1.C1Preview.RenderGraphics()

rg1.Graphics.FillEllipse(Brushes.DarkBlue,200,200,150,150)

rg1.Graphics.FillPie(Brushes.DarkRed,200,200, 150,150, -45, 75)

'添加 RenderObjects 至文档。Me.C1PrintDocument1.Body.Children.Add(rt1) Me.C1PrintDocument1.Body.Children.Add(ri1) Me.C1PrintDocument1.Body.Children.Add(rg1)

End Sub

* + C#

privatevoidMakeDoc()

{

//创建 RenderText。

C1.C1Preview.RenderTextrt1= new C1.C1Preview.RenderText(); rt1.Text= "Thisis RenderText.A RenderImagewillbe on page2 and

a RenderGraphicon page3.";

//添加一个分页符。

rt1.BreakAfter= C1.C1Preview.BreakEnum.Page;

//创建RenderImage。

C1.C1Preview.RenderImageri1= new C1.C1Preview.RenderImage();

ri1.Image= System.Drawing.Image.FromFile("c:\\c1logo.bmp");

//添加一个分页符。

ri1.BreakAfter= C1.C1Preview.BreakEnum.Page;

//创建一个RenderGraphic。

C1.C1Preview.RenderGraphicsrg1= new C1.C1Preview.RenderGraphics();

rg1.Graphics.FillEllipse(Brushes.DarkBlue,200,200,150,150);

rg1.Graphics.FillPie(Brushes.DarkRed,200,200,150,150,-45, 75);

//添加 RenderObjects 至文档。this.c1PrintDocument1.Body.Children.Add(rt1); this.c1PrintDocument1.Body.Children.Add(ri1); this.c1PrintDocument1.Body.Children.Add(rg1);

}

###### 您所达到的效果

不同的RenderObject之间均插入了一个分页符：



### 横向打印

C1PrintDocument 允许为文档指定不同的打印选项和页面设置。页面设置可以通过设计器或者代码指定。

###### 通过设计器

1. 在Form上选中**C1PrintDocument**。其属性将显示在VisualStudio右侧面板属性窗体上。
2. 找到C1PrintDocument 的**PageSettings**属性。



1. 找到**Landscape**属性并将其设置为**True**。则您的C1PrintDocument 将横向打印。



###### 通过代码

通过向**Form\_Load**事件添加以下代码，您可以在运行时修改C1PrintDocument的页面设置：

* + VisualBasic

Me.C1PrintDocument1.PageLayout.PageSettings.Landscape= True

* + C#

this.c1PrintDocument1.PageLayout.PageSettings.Landscape=true;

###### 运行时

此外，您还有机会在运行文档时，在C1PrintDocument的预览界面修改页面设置，通过页面设置对话框。

1. 单击位于工具栏的页面设置按钮。



弹出页面设置对话框。



页面设置对话框包含以下字段：

|  |  |
| --- | --- |
| **字段名称** | **描述** |
| Paper | 选择纸张尺寸和来源。不同的尺寸种类和来源依赖与选中的打印机。 |
| Orientation | 选择打印方向，纵向或者横向。 |
| Margins | 自定义C1PrintDocument的页边距（上下左右）。 |

页面设置对话框具有以下命令按钮：

|  |  |
| --- | --- |
| **按钮名称** | **描述** |
| **OK** | 应用设置到C1PrintDocument。 |
| **Cancel** | 取消对文档所做的页面设置的更改。 |
| **Printer** | 修改或查看打印机设置。 |

1. 找到**Orientation**字段并选择**Landscape**选项。您的C1PrintDocument将显示为横向，并将横向打印。



### 设置页面尺寸

通过PaperKind属性设置文档的页面尺寸。

1. 从工具栏上向您的工程添加C1PrintPreviewControl 和C1PrintDocument控件。
2. 单击C1PrintPreviewControl1 选中至属性窗体，设置其Document属性为

###### C1PrintDocument1。

1. 添加以下代码至Form\_Load事件：
	* VisualBasic

'构建文档。

MakeDoc()

'生成文档。

Me.C1PrintDocument1.Generate()

* + C#

//构建文档。

MakeDoc();

//生成文档。

this.c1PrintDocument1.Generate();

1. 添加以下MakeDoc子方法，通过PaperKind属性设置页面尺寸为**Legal**：
	* VisualBasic

PrivateSub MakeDoc()

'为文档定义页面布局。

Dim pl As New C1.C1Preview.PageLayout()

pl.PageSettings= New C1.C1Preview.C1PageSettings()

pl.PageSettings.PaperKind= System.Drawing.Printing.PaperKind.Legal Me.C1PrintDocument1.PageLayouts.Default= pl

End Sub

* + - C#

privatevoidMakeDoc()

{

//为文档定义页面布局。

C1.C1Preview.PageLayoutpl= new C1.C1Preview.PageLayout(); pl.PageSettings= new C1.C1Preview.C1PageSettings(); pl.PageSettings.PaperKind=

System.Drawing.Printing.PaperKind.Legal;

this.c1PrintDocument1.PageLayouts.Default= pl;

}

###### 您所达到的效果

默认的页面尺寸设置为**Legal**：



### 调整或者缩放图像大小

您可以方便的通过以下步骤将RenderImage缩放为原始尺寸的50%：

1. 从工具栏上向您的工程添加C1PrintPreviewControl 和C1PrintDocument组件。
2. 单击选中C1PrintPreviewControl1 ，在属性窗体设置Document属性为

###### C1PrintDocument1。

1. 切换至代码视图，添加以下命名空间声明：
	* VisualBasic

ImportsC1.C1Preview

* + C#

usingC1.C1Preview;

1. 添加以下**Form\_Load**事件处理，向页面添加一个图片并将其缩放至原始宽度的50%，高度按比例自动缩放，请将以下c1logo.png替换为自己的图片名称和位置：
	* VisualBasic

PrivateSub Form1\_Load(ByValsenderAs System.Object,ByVale As System.EventArgs)HandlesMyBase.Load

'创建一个新的RenderImage，使用您的图片名称和位置替换c1logo.png。

Dim img As New C1.C1Preview.RenderImage

img.Image= Image.FromFile("C:\c1logo.png")

'缩放图片为可用页面尺寸的50%，图片的高度将自动缩放。

img.Width= "50%"

'创建文档。

C1PrintDocument1.StartDoc() C1PrintDocument1.RenderBlock(img) C1PrintDocument1.EndDoc()

End Sub

* + C#

privatevoidForm1\_Load(objectsender,EventArgse)

{

//创建一个新的RenderImage，使用您的图片名称和位置替换c1logo.png。

C1.C1Preview.RenderImageimg= new C1.C1Preview.RenderImage(); img.Image= Image.FromFile("C:\\c1logo.png");

//缩放图片为可用页面尺寸的50%，图片的高度将自动缩放。

img.Width= "50%";

//创建文档。

c1PrintDocument1.StartDoc(); c1PrintDocument1.RenderBlock(img); c1PrintDocument1.EndDoc();

}

###### 您所达到的效果

页面上显示一幅缩放过的图片：



### 添加一个水印图片

通过Watermark属性向页面添加一幅水印图片。

1. 从工具栏上向您的工程添加**C1PrintPreviewControl**和**C1PrintDocument**组件。
2. 单击选中C1PrintPreviewControl1 ，在属性窗体设置Document属性为

###### C1PrintDocument1.

1. 切换至代码视图，添加以下命名空间声明：
	* VisualBasic

ImportsC1.C1Preview

* + C#

usingC1.C1Preview;

1. 添加以下**Form\_Load**事件处理，通过Watermark属性添加一个水印至页面，请使用自己的图片名和位置替代c1logo.png：
	* VisualBasic

PrivateSub Form1\_Load(ByValsenderAs System.Object,ByVale As System.EventArgs)Handles MyBase.Load

'创建水印和布局。

Dim waterMarkAs New RenderImage

Dim pl As New C1.C1Preview.PageLayout()

'设置水印图像；使用自己的图片替换c1logo.png。

waterMark.Image= Image.FromFile("c:\c1logo.png")

waterMark.Y= New Unit(2,UnitTypeEnum.Inch)

pl.Watermark= waterMark

Me.C1PrintDocument1.PageLayout= pl

'生成文档。

Me.C1PrintDocument1.Generate()

End Sub

* + - C#

privatevoidForm1\_Load(objectsender,EventArgse)

{

//创建水印和布局。

RenderImagewaterMark= new RenderImage(); C1.C1Preview.PageLayoutpl= new C1.C1Preview.PageLayout();

//设置水印图像；使用自己的图片替换c1logo.png。

waterMark.Image= Image.FromFile("c:\\c1logo.png"); waterMark.Y= new Unit(2,UnitTypeEnum.Inch); pl.Watermark= waterMark; this.c1PrintDocument1.PageLayout= pl;

//生成文档。

this.C1PrintDocument1.Generate();

}

###### 您所达到的效果

页面添加了一个水印：



### 设置初始的缩放模式

设置**ZoomMode**属性为**ActualSize**，**PageWidth**，**TextWidth**，**WholePage**（默认值），或者**Custom**以设置初始的缩放模式。该属性可以通过设计器或者代码进行设置。

###### 通过智能设计器

1. 单击**PreviewPane**浮动工具栏上的**Layout**按钮以打开Layout对话框。
2. 从**ZoomMode**下拉菜单选择PageWidth：



###### 通过属性窗体

1. 在属性窗体找到C1PrintPreviewControl的PreviewPane属性，并展开属性节点。
2. 设置**ZoomMode**属性的值为**PageWidth**。



###### 通过代码

在**Form\_Load**事件中添加以下代码以设置ZoomMode为**PageWidth**：

* VisualBasic Me.C1PrintPreviewControl1.PreviewPane.ZoomMode= C1.Win.C1Preview.ZoomModeEnum.PageWidth
* C# this.c1PrintPreviewControl1.PreviewPane.ZoomMode= C1.Win.C1Preview.ZoomModeEnum.PageWidth;

###### 您所达到的效果

当文档加载时，页面缩放至适合页面宽度的大小：



### 从上下文菜单中移除一个项

在运行时，默认情况下右键单击C1PreviewPane会出现一个上下文菜单。该ContextMenuStrip包含操作预览的全部设置，包括File，Zoom以及Text工具栏。您可以通过添加一个ContextMenuStrip创建自己的上下文菜单并指定给PreviewPane.ContextMenuStrip属性。同样您也可以通过添加新的菜单项或者删除已有的菜单项自定义现有的ContextMenuStrip。

以下示例从上下文菜单移除了标准的“Copy”菜单。完成以下步骤：

1. 在**Form\_Load**事件中向**C1PreviewPane**的ContextMenuStrip的**Opening**事件添加一个事件处理函数：
	* VisualBasic

AddHandlerPreviewPane.ContextMenuStrip.Opening,AddressOf ContextMenuStrip\_Opening

* + C#

PreviewPane.ContextMenuStrip.Opening+= new CancelEventHandler(ContextMenuStrip\_Opening);

1. 创建该**ContextMenuStrip\_Opening**事件处理，添加以下代码从上下文菜单移除“Copy”菜单项：
	* VisualBasic

PrivateSub ContextMenuStrip\_Opening(ByValsenderAs Object,ByVale As CancelEventArgs)

Dim cms As System.Windows.Forms.ContextMenuStrip= DirectCast(sender,System.Windows.Forms.ContextMenuStrip)

For EachitemAs ToolStripItemIn cms.Items

If item.Tag= ContextMenuTags.CopyThen item.Visible= False

End If

Next

End Sub

* + - C#

voidContextMenuStrip\_Opening(objectsender,CancelEventArgse)

{

System.Windows.Forms.ContextMenuStripcms= (System.Windows.Forms.ContextMenuStrip)sender;

foreach(ToolStripItemitemin cms.Items)

if (item.Tag== ContextMenuTags.Copy) item.Visible= false;

}

###### 您所达到的效果

当您右键单击C1PrintPreviewControl控件上的预览面板时，可以注意到上下文菜单上不包含标准的“Copy”菜单项。

### 禁用上下文菜单

当运行时通过右键单击一个C1PreviewPane时，会出现一个默认的上下文菜单。此上下文菜单包含操作预览的设置。您可以在代码中通过将C1PreviewPane的**ContextMenuStrip**属性设置为null禁用该上下文菜单。注意，这一操作无法通过设计器完成，只能通过代码做到。但是您可以在设计器中使用您自己的上下文菜单覆盖默认的菜单（通过在Form窗体上拖拽一个**ContextMenuStrip**组件，并设置C1PreviewPane的**ContextMenuStrip**至该组件）。当时用C1PrintPreviewControl控件时，预览面板可以通过控件上的PreviewPane属性进行访问。

在**Form\_Load**事件中添加以下代码以禁用C1PrintPreviewControl控件上的上下文菜单：

* VisualBasic

Me.C1PrintPreviewControl1.PreviewPane.ContextMenuStrip= Nothing

* C#

this.c1PrintPreviewControl1.PreviewPane.ContextMenuStrip= null;

###### 您所达到的效果

当您在C1PrintPreviewControl控件的预览面板右键单击时，不会出现默认的上下文菜单。