## 基于任务的Excel for .NET帮助文档

基于任务的帮助文档要求你具有使用VisualStudio.NET 进行编程的经验，完成本文档中的任务后，你就可以创建工程来展示 **Excelfor.NET**的特性，熟悉和了解**Excelfor.NET**的应用场景。

每一个基于任务的帮助话题都假定你已经创建了一个新的 .NET 项目，并且已经添加了合适的命名空间引用语句，比如 C# 中，使用 （**usingC1.C1Excel;**）， 而 Visual Basic则使用 （**ImportsC1.C1Excel**）。

参考

[添加内容到工作表](#_bookmark18)

[合并单元格](#_bookmark19)

[格式化单元格](#_bookmark20)

[在工作薄之间拷贝单元格行](#_bookmark21)

[添加图片到单元格](#_bookmark22)

[添加批注到单元格](#_bookmark23)

[添加分页符到单元表](#_bookmark24)

[为工作薄设定计算模式](#_bookmark25)

[导入和导出OpenXML文件](#_bookmark26)

[创建汇总结果](#_bookmark27)

[保存和加载 CSV 文件](#_bookmark28)

### 添加内容到工作表

按照下面的步骤将创建一个新的工作表，并且添加值到第一列的前十个单元格：

1. 在工具箱上双击[C1XLBook](#_bookmark36) 组件，将会添加该组件到表单上。
2. 添加值到第一列的前十个单元格

Visual Basic代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim sheet As XLSheet = C1XLBook1.Sheets(0) Dim i As IntegerFor i = 0 To 9sheet(i,0).Value = i + 1 Next i |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| XLSheet sheet = c1XLBook1.Sheets[0]; for (int i = 0; i <= 9; i++){sheet[i,0].Value = i + 1;} |

1. 保存文件，并且使用 Excel 打开该文件：

VisualBasic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| C1XLBook1.Save("c:\mybook.xls") System.Diagnostics.Process.Start("C:\mybook.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| c1XLBook1.Save(@"c:\mybook.xls"); System.Diagnostics.Process.Start(@"C:\mybook.xls"); |



### 合并单元格

通过[C1Excel 的 MergedCells](#_bookmark510) 属性可以合并指定区域的单元格，该属性通常被用来在工作表上检测，创建以及删除合并的单元格。每一个合并的单元格区域都会用一个[C1.C1Excel.XLCellRange](#_bookmark148) 对象来代表。添加删除单元格行或者单元格列，合并的单元格会被保留。

按照下面的步骤进行单元格合并：

1. 在工具箱上双击[C1XLBook](#_bookmark36) 组件，将会添加该组件到表单上。
2. 按照下面的代码，先创建两个单元格区域（\_**ColRange**和\_**RowRange**），然后添加这两个区域到MergedCells 中，最后给合并的单元格添加样式。你可以把这些代码写在**Form1\_Load**的事件处理函数中，代码如下：

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| private void Form1\_Load(object sender, EventArgs e){// 选择需要合并的单元格区域XLCellRange \_ColRange = new C1.C1Excel.XLCellRange(4, 6, 0, 8);XLCellRange \_RowRange = new C1.C1Excel.XLCellRange(9, 21, 3, 4);// 为合并的单元格设定文本c1XLBook1.Sheets[0][4, 0].Value = "Merged Cells"; c1XLBook1.Sheets[0][9, 3].Value = "Merged Cells";// 合并单元格c1XLBook1.Sheets[0].MergedCells.Add(\_ColRange); c1XLBook1.Sheets[0].MergedCells.Add(\_RowRange);// 定义样式，并且应用样式到合并单元格XLStyle \_Colstyle = new C1.C1Excel.XLStyle(c1XLBook1);\_Colstyle.BackColor = Color.Yellow; XLStyle \_Rowstyle = new XLStyle(c1XLBook1);\_Rowstyle.BackColor = Color.LightBlue; c1XLBook1.Sheets[0][4, 0].Style = \_Colstyle; c1XLBook1.Sheets[0][9, 3].Style = \_Rowstyle;// 保存并且打开工作薄c1XLBook1.Save(@"c:\mybook.xls"); System.Diagnostics.Process.Start(@"C:\mybook.xls");} |



### FormattingCells 格式化单元格区域

1. 在工具箱上双击[C1XLBook](#_bookmark36) 组件，将会添加该组件到表单上。
2. 创建一个新的样式，然后设定样式属性：

Visual Basic 代码

**VisualBasic**

Dim style1 As New XLStyle(C1XLBook1) style1.ForeColor = Color.Gold style1.BackColor = Color.Blue style1.Format = "$ .00"

C# 代码

**C#**

XLStyle style1 = new XLStyle(c1XLBook1); style1.ForeColor = Color.Gold; style1.BackColor = Color.Blue; style1.Format = "$ .00";

1. 给工作表的第一列单元格添加内容和样式：

Visual Basic 代码

**VisualBasic**

Dim sheet As XLSheet = C1XLBook1.Sheets(0) Dim i As Integer

For i = 0 To 9

sheet(i,0).Value = i + 1 sheet(i, 0).Style = style1

Next i

C# 代码

**C#**

C1.C1Excel.XLSheet sheet = c1XLBook1.Sheets[0]; int i;

for (i = 0; i <= 9; i++)

{

sheet[i,0].Value = i + 1;

sheet[i, 0].Style = style1;

}

1. 保存文件，然后使用 Excel 打开该文件。

Visual Basic 代码

**VisualBasic**

C1XLBook1.Save("c:\mybook.xls") System.Diagnostics.Process.Start("C:\mybook.xls")

C# 代码

**C#**

c1XLBook1.Save(@"c:\mybook.xls"); System.Diagnostics.Process.Start(@"C:\mybook.xls");



### 在工作薄之间拷贝单元格行

按照下面的步骤可以从一个工作薄中拷贝一些单元格行到另一个工作薄：

* 1. 加载一个现存的工作薄：

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim wb As New C1XLBook() wb.Load("C:\test.xls") |

C# 代码

C# 复制代码

C1XLBook wb = new C1XLBook(); wb.Load(@"C:\test.xls");

* 1. 创建一个新的工作表（XLSheet）对象

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim xb As New C1XLBook() xb.Sheets.Add("Test") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| C1XLBook xb = new C1XLBook(); xb.Sheets.Add("Test"); |

* 1. 逐行从现有的工作薄中拷贝数据，然后复制到新的工作表：

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim source As XLSheet = wb.Sheets(0) Dim dest As XLSheet = xb.Sheets("Test") Dim row As Integer, col As IntegerFor row = 0 To source.Rows.Count - 1For col = 0 To source.Columns.Count - 1 dest(row, col).Value = source(row, col).ValueNext col Next row |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| XLSheet source = wb.Sheets[0]; XLSheet dest = xb.Sheets("Test");for (int row = 0; row <= source.Rows.Count - 1; row++){for (int col = 0; col <= source.Columns.Count - 1; col++){dest[row, col].Value = source[row, col].Value;}} |

* 1. 保存文件，使用 Excel 打开新创建的文件：

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| xb.Save("C:\test2.xls") System.Diagnostics.Process.Start("C:\test2.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| xb.Save(@"c:\test2.xls"); System.Diagnostics.Process.Start(@"C:\test2.xls"); |

### 添加图片到单元格

按照下面的方法可以添加图片到单元格或者工作表，点击下面的链接可以看见详细的步骤：

方法1: 直接给单元格的 Value 属性赋值一个图片对象。

按照这个方法添加到工作表的图片会保持原始尺寸，图片的左上角会对齐单元格的左上角。

1. 加载一个现存的工作薄，创建一个新的工作表，并且添加内容。

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim wb As New C1XLBook wb.Load("C:\Project\WorkBook1.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| C1XLBook wb = new C1XLBook(); wb.Load(@"C:\Project\WorkBook1.xls"); |

1. 加载图片，并且赋给单元格的[Value](#_bookmark147)属性。

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim img As Image = Image.FromFile("C:\Project\MyImage.bmp") Dim sheet As XLSheet = wb.Sheets("Forecasting Report") sheet(0, 0).Value = img |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| Image img = Image.FromFile(@"C:\Project\MyImage.bmp"); XLSheet sheet = wb.Sheets("Forecasting Report"); sheet[0,0].Value = img; |

1. 保存文件，然后使用 Excel 打开该文件：

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| wb.Save("C:\Project\WorkBook1.xls ") System.Diagnostics.Process.Start("C:\Project\WorkBook1.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| wb.Save(@"C:\Project\WorkBook1.xls"); System.Diagnostics.Process.Start(@"C:\Project\WorkBook1.xls"); |

在这个例子中图片会作为第一个单元格的值，并且保持其原有尺寸出现在单元格中。



方法 2: 创建一个新的XLPictureShape 对象，设置属性，然后将该对象设置为单元格的 Value 属性。

使用这个方法添加的图片，开发人员可以改变大小，旋转角度，透明度，对比度以及边框等等。

1. 加载现有的工作薄，添加一个新的工作表，并且增加一些内容。

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim wb As New C1XLBook wb.Load("C:\Project\WorkBook1.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| C1XLBook wb = new C1XLBook(); wb.Load(@"C:\Project\WorkBook1.xls"); |

1. 创建一个[XLPictureShape](#_bookmark347)对象，更改属性，然后赋给单元格的值属性。

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim img As Image = Image.FromFile("C:\Project\MyImage.bmp") Dim pic As New XLPictureShape(img, 1500, 1500) pic.Rotation = 30.0Fpic.LineColor = Color.DarkRed pic.LineWidth = 100'设定图片到工作表的第一个单元格Dim sheet As XLSheet = wb.Sheets("Forecasting Report") sheet(0, 0).Value = pic |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| Image img = Image.FromFile(imageFileName);XLPictureShape pic = new XLPictureShape(img, 1500, 1500); pic.Rotation = 30.0f; |

pic.LineColor = Color.DarkRed; pic.LineWidth = 100;

//设定图片到工作表的第一个单元格

XLSheet sheet = wb.Sheets("Forecasting Report"); sheet[0,0].Value = pic;

1. 保存工作薄，然后使用 Excel 打开：

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| wb.Save("C:\Project\WorkBook1.xls ") System.Diagnostics.Process.Start("C:\Project\WorkBook1.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| wb.Save(@"C:\Project\WorkBook1.xls"); System.Diagnostics.Process.Start(@"C:\Project\WorkBook1.xls"); |

在这个例子中图片将会作为第一个单元格的值，然后旋转 30°，添加深红色的边框。 由于我们已经指定了这个图片的位置，它并不会出现在第一个单元格中。



方法3: 创建一个XLPictureShape 对象，设定其属性，然后添加该对象到工作表的ShapeCollection 集合中。

这个方法使用XLPictureShape 对象的构造器来指定图片的边界，这个图形对象被直接添加到工作表的ShapeCollection 集合中，而不是绑定到单元格上面。

1. 加载现有的工作薄，添加一个新的工作表，并且增加一些内容。

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim wb As New C1XLBook wb.Load("C:\Project\WorkBook1.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| C1XLBook wb = new C1XLBook(); wb.Load(@"C:\Project\WorkBook1.xls"); |

1. 创建一个[XLPictureShape](#_bookmark347)对象，更改属性，然后添加到工作表的 ShapeCollection 集合中去

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| Dim img As Image = Image.FromFile("C:\Project\MyImage.bmp") Dim pic As New XLPictureShape(img, 3000, 3500, 2500, 900) pic.Rotation = 30.0Fpic.LineColor = Color.DarkRedpic.LineWidth = 100'添加图片到工作表的 Shapes 集合Dim sheet As XLSheet = wb.Sheets("Forecasting Report") sheet.Shapes.Add(pic) |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| Image img = Image.FromFile(@"C:\Project\MyImage.bmp"); XLPictureShape pic = new XLPictureShape(img, 3000, 3500, 2500, 900); pic.Rotation = 30.0f;pic.LineColor = Color.DarkRed; pic.LineWidth = 100;//添加图片到工作表的 Shapes 集合XLSheet sheet = wb.Sheets("Forecasting Report"); sheet.Shapes.Add(pic) |

1. 保存工作薄，然后使用 Excel 打开：

Visual Basic 代码

|  |  |
| --- | --- |
| VisualBasic | 复制代码 |
| wb.Save("C:\Project\WorkBook1.xls ") System.Diagnostics.Process.Start("C:\Project\WorkBook1.xls") |

C# 代码

|  |  |
| --- | --- |
| C# | 复制代码 |
| wb.Save(@"C:\Project\WorkBook1.xls"); System.Diagnostics.Process.Start(@"C:\Project\WorkBook1.xls"); |

在这个例子中创建的图形对象会被添加到工作表的[ShapeCollection](#_bookmark73)集合中，而第一个单元格的值并不会改变。然后我们指定图形对象的高度，宽度以及水平和竖直位置。



### 添加批注到单元格

1. 在工具箱上双击[C1XLBook](#_bookmark36) 组件，将会添加该组件到表单上。
2. 按照下面的代码给指定单元格添加文本：

Visual Basic 代码

**VisualBasic**

C1XLBook1.Sheets(0)(2, 3).Value = "test"

C# 代码

**C#**

c1XLBook1.Sheets[0][2,3].Value="test";

1. 添加批注到工作表的[XLCommentCollection](#_bookmark263)集合，然后构造一个矩形区域来显示批注：

Visual Basic 代码

**Visual Basic**

C1XLBook1.Sheets(0).Comments.Add(2, 3, "John", "Test comment") C1XLBook1.Sheets(0).Comments(0).TextBox.Rectangle = New Rectangle(220, 210, 1900,

1200)

C# 代码

**C#**

c1XLBook1.Sheets[0].Comments.Add(2, 3, "John", "Test comment"); c1XLBook1.Sheets[0].Comments[0].TextBox.Rectangle = new Rectangle(220, 210, 1900,

1200);

1. 保存文件，然后使用 Excel 打开：

Visual Basic 代码

**Visual Basic 复制代码**

C1XLBook1.Save("c:\mybook.xls") System.Diagnostics.Process.Start("C:\mybook.xls")

C# 代码

**C# 复制代码**

c1XLBook1.Save(@"c:\mybook.xls"); System.Diagnostics.Process.Start(@"C:\mybook.xls");

5.

Runtheprogram.Thespreadsheetwillopenandlooksimilartothefollowing: