[**.NET平台开源项目速览(16)C#写PDF文件类库PDF File Writer介绍**](http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html)

**阅读目录**

* [1.PDF File Writer基本介绍](http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html#_label0)
* [2.一个简单的使用案例](http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html#_label1)
* [3.资源](http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html#_label2)

    1年前，我在文章：[这些.NET开源项目你知道吗？.NET平台开源文档与报表处理组件集合(三)](http://www.cnblogs.com/asxinyu/p/dotnet_opensource_project_3.html)中(第9个项目)，给大家推荐了一个开源免费的PDF读写组件 PDFSharp，PDFSharp我2年前就看过，用过简单的例子，不过代码没有写成专门的文章。最近在查找资料的时候，又发现一款小巧的写PDF文件的C#组件：PDF File Writer。该开源组件是在codeproject，还没有托管到其他地方，所以花了点时间了解了一下，分享给大家。

    .NET开源目录：[【目录】本博客其他.NET开源项目文章目录](http://www.cnblogs.com/asxinyu/p/Bolg_Category_For_DotNet_OpenSource.html)

本文原文地址：<http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html> 

[回到目录](http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html#_labelTop)

**1.PDF File Writer基本介绍**

**1.1 支持的常用功能**

[PDF File Writer](http://www.codeproject.com/Articles/570682/PDF-File-Writer-Csharp-Class-Library-Version)组件可以在.NET应用程序中直接创建PDF格式的文件。最低的开发环境是.NET 4.0 + VS 2013。我们来看看该组件支持的PDF的一些功能：

　　图形:支持画线、矩形、多边形、贝塞尔曲线,前景和背景颜色,模式和阴影。

　　图片:支持位图图像和矢量图像

   　文本:支持行文本和列文本

　　条形码:支持条形码：Barcode 128, Barcode 39, Barcode interleaved 2 of 5等

　　二维码:支持二维条码

       加密：支持AES-128加密算法

       Web链接:支持Web超链接

       书签：支持文档大纲

       图表：支持微软的图表，支持数据表格，支持声音，视频播放；

**1.2 使用PDF File Writer创建PDF的步骤**

使用PDF File Writer在程序中创建一个PDF文件的主要步骤如下：

    Step 1: 创建PdfDocument文件对象

    Step 2: 创建资源对象，如文字(PdfFont)，图像(PdfImage)等

    Step 3: 创建文件页对象PdfPage

    Step 4: 创建内容对象PdfContents

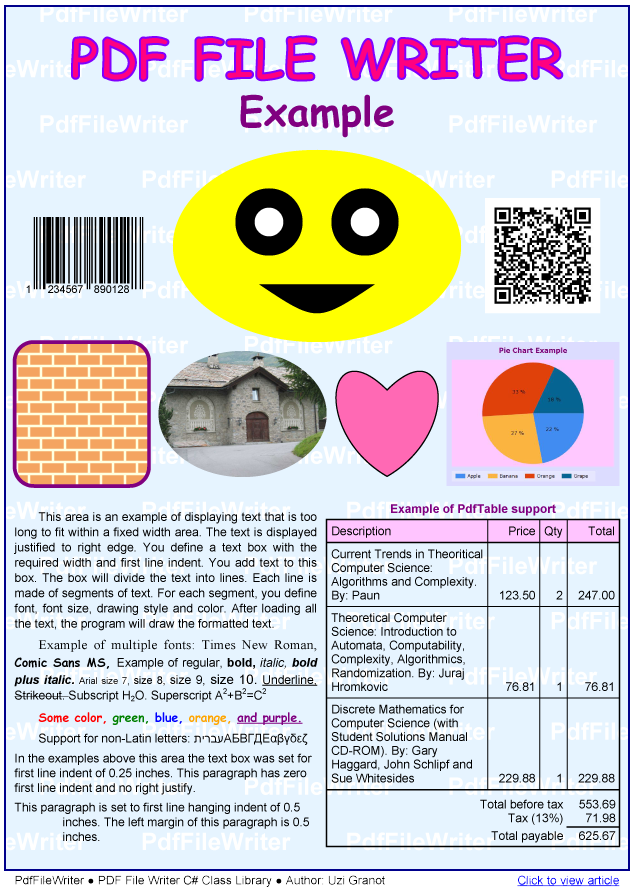
    Step 5: 在内容对象上添加文字，或者图像等内容

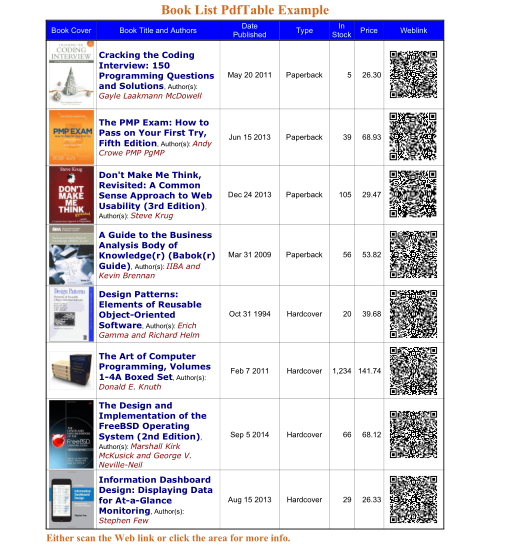
    重复3, 4 ，5 创建其他页

    Step 6: 使用PdfDocument对象的CreateFile方法创建PDF文

**1.3 PDF File Writer创建的PDF文件效果预览**

    看看使用PDF File Writer创建的PDF的效果，非常不错。这也是我偶尔碰到非常震撼，拿过来分享的重要原因。





[回到目录](http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html#_labelTop)

**2.一个简单的使用案例**

我们根据官方提供的例子，可以快速入门，一起来看看基本代码。

**2.1 先创建基本对象**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | private PdfFont            ArialNormal;  private PdfFont            ArialBold;  private PdfFont            ArialItalic;  private PdfFont            ArialBoldItalic;  private PdfFont            TimesNormal;  private PdfFont            Comic;  private PdfTilingPattern WaterMark;  private PdfDocument        Document;  private PdfPage            Page;  private PdfContents        Contents; |

    然后创建空白文档

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | // Step 1:创建空文档，文档参数有类型，可以使用枚举进行选择，和返回的文件名称  Document = new PdfDocument(PaperType.Letter, false, UnitOfMeasure.Inch, FileName);  //加密测试例子  //Document.SetEncryption(null, null, Permission.All & ~Permission.Print, EncryptionType.Aes128);  //创建PDF文件信息目录  PdfInfo Info = PdfInfo.CreatePdfInfo(Document);  Info.Title("Article Example");  Info.Author("Uzi Granot Granotech Limited");  Info.Keywords("PDF, .NET, C#, Library, Document Creator");  Info.Subject("PDF File Writer C# Class Library (Version 1.14.1)"); |

**2.2 创建字体等资源**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | //定义不同的字体类型，如下所示  String FontName1 = "Arial";  String FontName2 = "Times New Roman";    ArialNormal = PdfFont.CreatePdfFont(Document, FontName1, FontStyle.Regular, true);  ArialBold = PdfFont.CreatePdfFont(Document, FontName1, FontStyle.Bold, true);  ArialItalic = PdfFont.CreatePdfFont(Document, FontName1, FontStyle.Italic, true);  ArialBoldItalic = PdfFont.CreatePdfFont(Document, FontName1, FontStyle.Bold | FontStyle.Italic, true);  TimesNormal = PdfFont.CreatePdfFont(Document, FontName2, FontStyle.Regular, true);  Comic = PdfFont.CreatePdfFont(Document, "Comic Sans MS", FontStyle.Bold, true); |

**2.3 创建文字示例**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | Contents.DrawText(Comic, 40.0, 4.25, 9.25, TextJustify.Center, 0.02, Color.FromArgb(128, 0, 255), Color.FromArgb(255, 0, 128), "PDF FILE WRITER");  Contents.SaveGraphicsState();  Contents.SetColorNonStroking(Color.Purple);  Contents.DrawText(Comic, 30.0, 4.25, 8.75, TextJustify.Center, "Example");  Contents.RestoreGraphicsState();  //Step 3:添加新页面  Page = new PdfPage(Document);  //Step 4:添加内容到页面  Contents = new PdfContents(Page); |

**2.4 绘制条形码**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | Contents.SaveGraphicsState();  BarcodeEAN13 Barcode1 = new BarcodeEAN13("1234567890128");  Contents.DrawBarcode(1.3, 7.05, 0.012, 0.75, Barcode1, ArialNormal, 8.0);  PdfQRCode QRCode = new PdfQRCode(Document, "http://www.codeproject.com/Articles/570682/PDF-File-Writer-Csharp-Class-Library-Version", ErrorCorrection.M);  Contents.DrawQRCode(QRCode, 6.0, 6.8, 1.2);  // 添加链接  Page.AddWebLink(6.0, 6.8, 7.2, 8.0, "http://www.codeproject.com/Articles/570682/PDF-File-Writer-Csharp-Class-Library-Version");  //保存  Contents.RestoreGraphicsState(); |

**2.5 绘制图表**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | Contents.SaveGraphicsState();    //创建MS Chart图表  Chart PieChart = PdfChart.CreateChart(Document, 1.8, 1.5, 300.0);  PdfImageControl ImageControl = new PdfImageControl();  ImageControl.SaveAs = SaveImageAs.IndexedImage;  PdfChart PiePdfChart = new PdfChart(Document, PieChart, ImageControl);    PieChart.AntiAliasing = AntiAliasingStyles.None;    //设置颜色  PieChart.BackColor = Color.FromArgb(220, 220, 255);  PieChart.Palette = ChartColorPalette.BrightPastel;    //默认字体  Font DefaultFont = PiePdfChart.CreateFont("Verdana", FontStyle.Regular, 0.05, FontSizeUnit.UserUnit);  Font TitleFont = PiePdfChart.CreateFont("Verdana", FontStyle.Bold, 0.07, FontSizeUnit.UserUnit);    // 设置标题  Title Title1 = new Title("Pie Chart Example", Docking.Top, TitleFont, Color.Purple);  PieChart.Titles.Add(Title1);    //图例  Legend Legend1 = new Legend();  PieChart.Legends.Add(Legend1);  Legend1.BackColor = Color.FromArgb(230, 230, 255);  Legend1.Docking = Docking.Bottom;  Legend1.Font = DefaultFont;    // 图表区域  ChartArea ChartArea1 = new ChartArea();  PieChart.ChartAreas.Add(ChartArea1);    ChartArea1.BackColor = Color.FromArgb(255, 200, 255);    Series Series1 = new Series();  PieChart.Series.Add(Series1);  Series1.ChartType = SeriesChartType.Pie;  Series1.Font = DefaultFont;  Series1.IsValueShownAsLabel = true;  Series1.LabelFormat = "{0} %";  Series1.Points.Add(22.0);  Series1.Points[0].LegendText = "Apple";  Series1.Points.Add(27.0);  Series1.Points[1].LegendText = "Banana";  Series1.Points.Add(33.0);  Series1.Points[2].LegendText = "Orange";  Series1.Points.Add(18.0);  Series1.Points[3].LegendText = "Grape";    Contents.DrawChart(PiePdfChart, 5.6, 5.0);  // 保存  Contents.RestoreGraphicsState(); |

**2.6 生成PDF**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | // Step 6:创建PDF  Document.CreateFile();  //打开PDF文件  Process Proc = new Process();  Proc.StartInfo = new ProcessStartInfo(FileName);  Proc.Start(); |

[回到目录](http://www.cnblogs.com/asxinyu/p/dotnet_Opensource_project_PdfFileWriter.html#_labelTop)

**3.资源**

1.Codeproject文章连接：<http://www.codeproject.com/Articles/570682/PDF-File-Writer-Csharp-Class-Library-Version>

2.PDF File Writer DLL下载：[PdfFileWriter\_dll.zip](http://files.cnblogs.com/files/asxinyu/PdfFileWriter_dll.zip)

3.PDF File Writer 帮助文档：[PdfFileWriterCHM.rar](http://files.cnblogs.com/files/asxinyu/PdfFileWriterCHM.rar)

4.PDF File Writer源代码与Demo:[PdfFileWriter-Code.rar](http://files.cnblogs.com/files/asxinyu/PdfFileWriter-Code.rar)

注意：源代码中的相关素材进行了精简，否则文件比较大，长传比较大。如果有需求可以去文章链接原文下载，或者单独留下邮箱，我有空发送一下。

如果您觉得阅读本文对您有帮助，请点一下“**推荐**”按钮，您的**“推荐”**将是我最大的写作动力！欢迎各位转载，但是未经作者本人同意，转载文章之后**必须在文章页面明显位置给出作者和原文连接**，否则保留追究法律责任的权利。