

## 使用ASP.NET (C#) 產生PDF檔的好幫手—iTextSharp library (下)

作者：唐瑤瑤 / 臺灣大學計算機及資訊網路中心程式設計組程式設計師

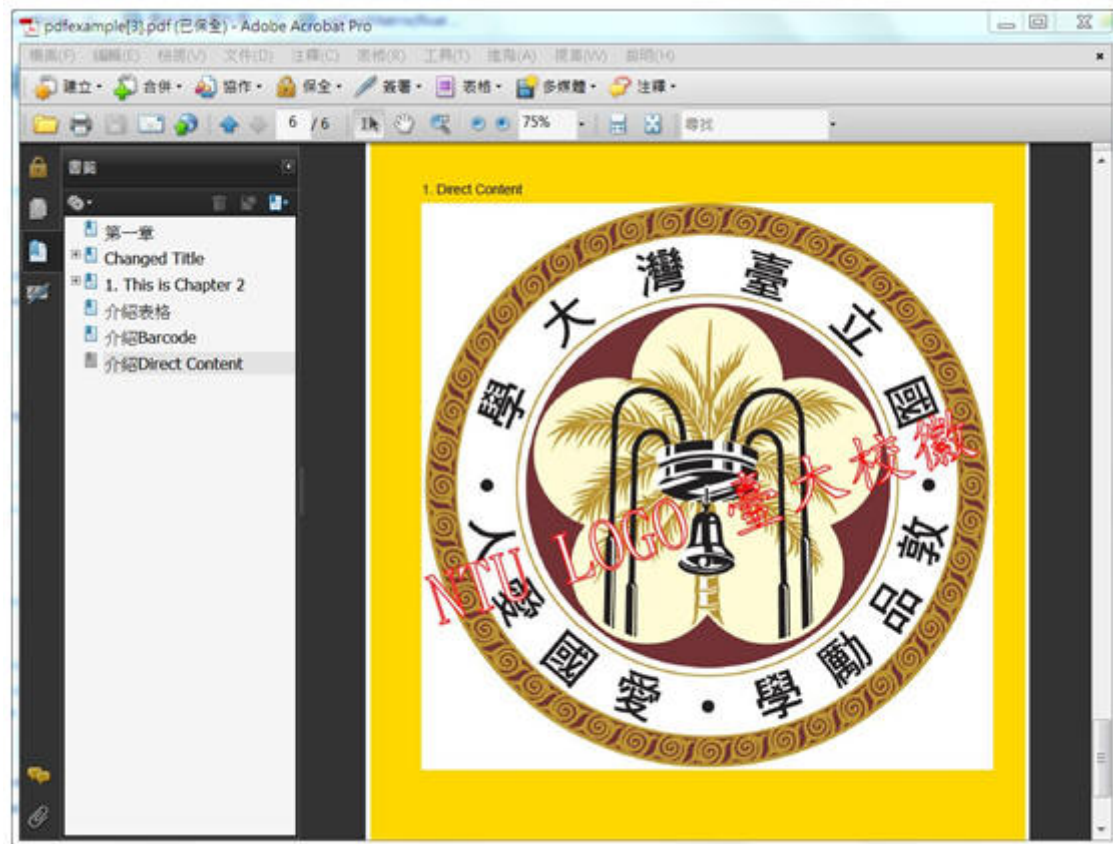
上一期的計中電子報中，我們向各位介紹了iTextSharp 這個十分強大且免費的PDF函式庫，相信各位對它應該已經不陌生了。接下來我們就針對幾項進階功能做介紹，包含Direct content、Header、Footer、Barcode及JavaScript等互動功能。

### Part 1：Direct content

Direct content 是指在頁面的絕對位置中插入內容。範例中會使用到的是PdfContentByte及ColumnText 2個類別。首先於頁面中加入一張圖檔，接著我們在圖片的【上方 upper layer】寫字，在圖片的【下方 lower layer】畫一個矩形並填滿色彩。主要使用到的方法是PdfWriter.DirectContent 及PdfWriter.DirectContentUnder：

```
// Add text on top of the image
PdfContentByte over = PdfWriter.DirectContent;
over.SaveState();
float sinus = (float) Math.Sin(Math.PI / 10);
float cosinus = (float) Math.Cos(Math.PI / 10);
BaseFont bf = BaseFont.CreateFont();
over.BeginText();
over.SetTextRenderingMode(PdfContentByte.TEXT_RENDER_MODE_FILL_STROKE);
over.SetLineWidth(1.5f);
over.SetRGBColorStroke(0xFF, 0x00, 0x00);
over.SetRGBColorFill(0xFF, 0xFF, 0xFF);
over.SetFontAndSize(bf, 60);
over.SetTextMatrix(cosinus, sinus, -sinus, cosinus, 65, 324);
over.ShowText("NTU LOGO 臺大校徽");
over.EndText();
over.RestoreState();
// Add a rectangle under the image
PdfContentByte under = PdfWriter.DirectContentUnder;
under.SaveState();
// 鮮黃色底
under.SetRGBColorFill(0xFF, 0xD7, 0x00);
// 周圍留白邊
under.Rectangle(5, 5, PageSize.A4.Width - 10, PageSize.A4.Height - 10);
under.Fill();
under.RestoreState();
```

產生的結果如下圖：



接下來我們在頁面絕對座標位置上任意畫幾條直線：



```
// Drawing lines
----- PdfContentByte canvas = PdfWriter.DirectContent;
----- canvas.SaveState();
----- canvas.SetLineWidth(0.05f);
----- canvas.MoveTo(400, -806 - 400);
----- canvas.LineTo(400, -626);
----- canvas.MoveTo(508.7f, -806);
----- canvas.LineTo(508.7f, -626);
----- canvas.MoveTo(280, -788);
----- canvas.LineTo(520, -788);
----- canvas.MoveTo(280, -752);
----- canvas.LineTo(520, -752);
----- canvas.MoveTo(280, -716);
----- canvas.LineTo(520, -716);
----- canvas.MoveTo(280, -680);
----- canvas.LineTo(520, -680);
----- canvas.MoveTo(280, -644);
----- canvas.LineTo(520, -644);
----- canvas.Stroke();
----- canvas.RestoreState();
```

如果是利用ColumnText 顯示文字內容於頁面絕對位置，請參考下列程式碼：

```
// Adding text with ColumnText.showTextAligned()  
String fooBar = "國立臺灣大學校徽";  
Phrase phrase = new Phrase(fooBar, ChFont);  
Phrase phrase1 = new Phrase("右下角", ChFont);  
ColumnText.ShowTextAligned(canvas, Element.ALIGN_LEFT, phrase, 200, 100,  
0); // x, y, rotation  
ColumnText.ShowTextAligned(canvas, Element.ALIGN_LEFT, phrase1,  
PageSize.A4.Right - 50, PageSize.A4.Left + 20, -30);
```

產生的結果如下圖：



## Part 2：Header及Footer

如果我們希望針對每一頁都能加上Header或Footer時，除了需要用到Part 1 提到的Direct Content，最重要是要處理Page Event，也就是PdfPageEventHelper介面。

可以override 事件如下：OnChapter、OnChapterEnd、OnCloseDocument、OnEndPage、OnGenericTag、OnOpenDocument、OnParagraph、OnParagraphEnd、OnSection、OnSectionEnd、OnStartPage。我們可以針對程式需求，挑出需要使用到的事件來改寫。

在我們的範例程式中，希望能於頁尾顯示頁數及檔案建立時間。所以我們先建立Page Event class，並改寫OnOpenDocument 方法。PdfTemplate 是在每一頁頁尾用來存放總頁數的值：

```
public class TwoColumnHeaderFooter : PdfPageEventHelper
{
    // This is the contentbyte object of the writer
    PdfContentByte cb;

    // we will put the final number of pages in a template
    PdfTemplate template;

    // this is the BaseFont we are going to use for the header / footer
    BaseFont bf = null;

    // This keeps track of the creation time
    DateTime PrintTime = DateTime.Now;

    // we override the onOpenDocument method
    public override void OnOpenDocument(PdfWriter writer, Document document)
    {
        try
        {
            PrintTime = DateTime.Now;

            bf = BaseFont.CreateFont(BaseFont.HELVETICA, BaseFont.CP1252, BaseFont.NOT_EMBEDDED);

            cb = writer.DirectContent;

            template = cb.CreateTemplate(50, 50);

        }
        catch (DocumentException de) {}
        catch (System.IO.IOException ioe) {}
    }
}
```



接著override `OnEndPage` 方法。我們將在每一頁右上角Header顯示”NTU C&INC”文字；左下角Footer顯示頁數與總頁數、右下角Footer 顯示文件產生的日期與時間。使用 `ShowTextAligned`、`ShowTextMatrix`、`ShowText` 等方法設定文字顯示的絕對位置，並保留好 `template` 的位置以顯示總頁數：



```
public override void OnEndPage(PdfWriter writer, Document document) +  
  
    { +  
  
        base.OnEndPage(writer, document); +  
  
        int pageN = writer.PageNumber; +  
  
        String text = "Page " + pageN + " of "; +  
  
        float len = bf.GetWidthPoint(text, 8); +  
  
        Rectangle pageSize = document.PageSize; +  
  
        cb.SetRGBColorFill(100, 100, 100); +  
  
        cb.BeginText(); +  
  
        cb.SetFontAndSize(bf, 8); +  
  
        cb.ShowTextAligned(PdfContentByte.ALIGN_RIGHT, +  
  
            "NTU-C&INC", +  
  
            pageSize.GetRight(40), +  
  
            pageSize.GetTop(30), 0); +  
  
        +  
  
        cb.SetTextMatrix(pageSize.GetLeft(40), +  
        pageSize.GetBottom(30)); +  
  
        cb.ShowText(text); +  
  
        cb.EndText(); +  
  
        cb.AddTemplate(template, pageSize.GetLeft(40) + len, +  
        pageSize.GetBottom(30)); +  
  
        cb.BeginText(); +  
  
        cb.SetFontAndSize(bf, 8); +  
  
        cb.ShowTextAligned(PdfContentByte.ALIGN_RIGHT, +  
  
            "Printed On " + PrintTime.ToString(), +  
  
            pageSize.GetRight(40), +  
  
            pageSize.GetBottom(30), 0); +  
  
        cb.EndText(); +  
  
    } +
```

```
public override void OnCloseDocument(PdfWriter writer, Document document) {  
    {  
        base.OnCloseDocument(writer, document);  
        template.BeginText();  
        template.SetFontAndSize(bf, 8);  
        template.SetTextMatrix(0, 0);  
        template.ShowText("" + (writer.PageNumber - 1));  
        template.EndText();  
    }  
}
```

The screenshot shows the Adobe Acrobat Pro interface. The top menu bar includes options like '檔案(F)', '編輯(E)', '檢視(V)', '文件(F)', '注釋(C)', '表格(T)', '工具(T)', '進階(A)', '版面(W)', and '窗口(W)'. Below the menu is a toolbar with icons for '建立', '合併', '協作', '安全', '簽署', '表格', '多媒體', and '注釋'. The main window displays a PDF document titled 'pdfexample[1].pdf (已保全)'. The left sidebar shows a table of contents with items like '第一章', 'Changed Title', '1. This is Chapter 2', '介紹表格', '介紹Barcode', '介紹Direct Content', and 'Content'. The main content area shows a page with a header section containing a table with four cells labeled 'Cell 1', 'Cell 2', 'Cell 3', and 'Cell 4'. The page number 'Page 3 of 6' is visible in the top right corner. The status bar at the bottom indicates '8.26 x 11.69 英寸'.

iTextSharp 提供直覺的方式來產生各種條碼，包括Barcode、QR Code 等。您也可以利用上一

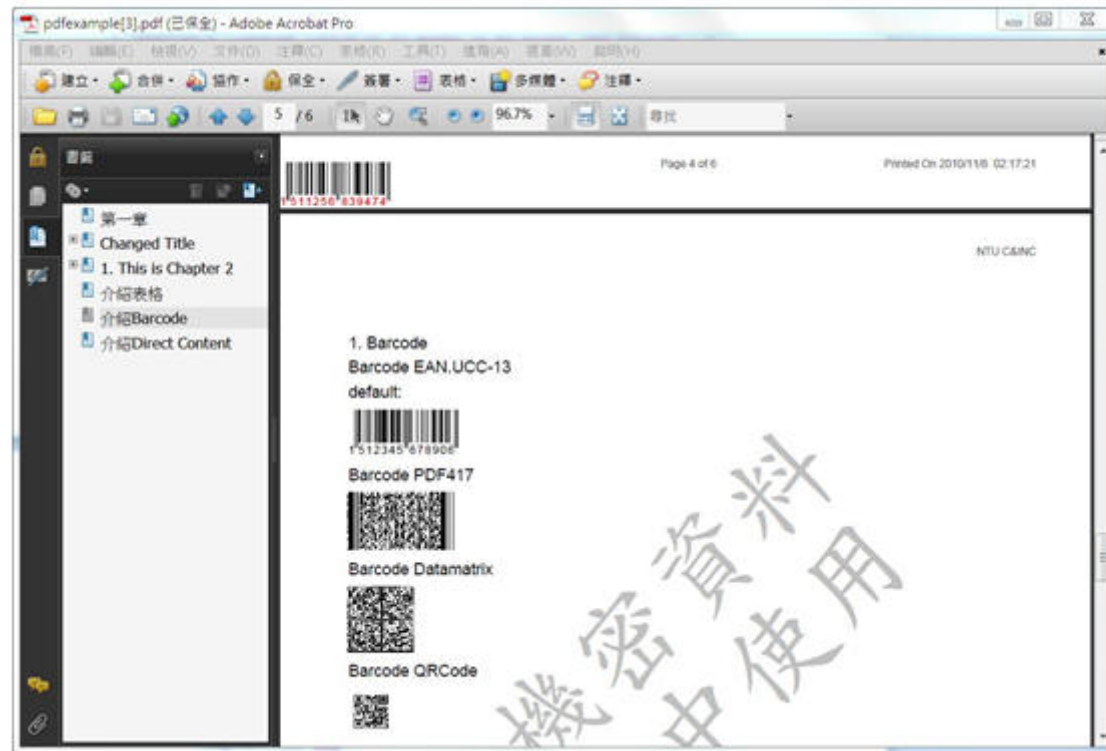
節的**Page event** 作法於每一頁頁尾產生文件條碼，使用**PlaceBarcode** 方法即可。程式碼很容易懂，直接複製如下：

```
//Place Barcode
-----PdfContentByte cb =PdfWriter.DirectContent;-----
-----// EAN-13
-----chapter4.Add(new Paragraph("Barcode EAN.UCC-13"));
-----BarcodeEAN codeEAN = new BarcodeEAN();
-----codeEAN.Code = "1512345678906";
-----chapter4.Add(new Paragraph("default:"));
-----chapter4.Add(codeEAN.CreateImageWithBarcode(cb, null, null));
-----// PDF417
-----chapter4.Add(new Paragraph("Barcode PDF417"));
-----BarcodePDF417 pdf417 = new BarcodePDF417();

String text = "Call me Ishmael. Some years ago--never mind how long
+ "precisely --having little or no money in my purse, and nothing
"particular to interest me on shore, I thought I would sail about
+ "a little and see the watery part of the world.";
-----pdf417.SetText(text);
-----iTextSharp.text.Image img = pdf417.GetImage();
-----img.ScalePercent(50, 50 * pdf417.YHeight);
-----chapter4.Add(img);
-----chapter4.Add(new Paragraph("Barcode Datamatrix"));
-----BarcodeDatamatrix datamatrix = new BarcodeDatamatrix();
-----datamatrix.Generate(text);
-----img = datamatrix.CreateImage();
-----chapter4.Add(img);
-----chapter4.Add(new Paragraph("Barcode QRCode"));
-----BarcodeQRCode qrcode = new BarcodeQRCode("Moby Dick by Herman
Melville", 1, 1, null);
-----img = qrcode.GetImage();
-----chapter4.Add(img);
```



產生結果如下圖：



#### Part 4：利用javascript與使用者互動

在這裡我們使用PdfStamper 類別及Utilities 類別讀進javascript 檔的程式碼，

```
stamper.JavaScript = Utilities.ReadFileToString(printjs);
```

接著使用PdfAction 指定欲執行的javascript，

```
PdfAction action = PdfAction.JavaScript("app.alert('Think before you print!',1);", stamper.Writer);
```

```
action.Next(PdfAction.JavaScript("printCurrentPage(this.pageNum);", stamper.Writer));
```

最後將頁面上已設定好的執行區塊chunk 做SetAction 動作：

```
chunk.SetAction(action);
```

完整程式碼如下：

```
void PrintPDF(string fname) ↵  
  
    ...{ ↵  
  
        .....string path = Server.MapPath("PDF"); ↵  
  
        .....string printjs = Server.MapPath("js") + "/print_page.js"; ↵  
  
        .....PdfReader reader = new PdfReader(path + "/" + fname + ".pdf"); ↵  
  
        .....string FontPath = Server.MapPath("files/kaiu.ttf"); ↵  
  
        .....BaseFont bfChinese = BaseFont.CreateFont(FontPath, ↵  
BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED); ↵  
  
        .....Font chfont = new Font(bfChinese, 12); ↵  
↵  
  
        .....using (MemoryStream memoryStream = new MemoryStream()) ↵  
  
        .....{ ..... ↵  
  
            .....// Create the stamper ↵  
  
            .....PdfStamper stamper = new PdfStamper(reader, memoryStream); ↵  
  
            .....Rectangle pageSize = PageSize.A4; ↵  
  
            .....// Add JavaScript ↵  
  
            .....stamper.JavaScript = Utilities.ReadFileToString(printjs); ↵  
  
            .....// Create a Chunk with a chained action ↵  
  
            .....PdfContentByte canvas; ..... ↵  
  
            .....Chunk chunk = new Chunk("列印本頁", chfont); ↵  
  
            .....PdfAction action = PdfAction.JavaScript("app.alert('Think ↵  
before you print!',1);", stamper.Writer); ↵
```

```

.....

action.Next(PdfAction.JavaScript("printCurrentPage(this.pageNum);",
stamper.Writer));

.....

..... chunk.SetAction(action);

..... Phrase phrase = new Phrase(chunk);

..... // Add this Chunk to every page .....

..... for (int i = 1; i <= reader.NumberOfPages; i++) // Must start at
1 because 0 is not an actual page.

..... {

..... canvas = stamper.GetOverContent(i);

..... ColumnText.ShowTextAligned(canvas,
PdfContentByte.ALIGN_RIGHT, phrase, pageSize.GetRight(40),
pageSize.GetTop(30), 0);

..... }

..... stamper.FormFlattening = true; // enable this if you want the PDF
flattened.

..... stamper.Close(); // Always close the stamper or you'll have a 0
byte stream.

..... reader.Close();

..... }

..... }


```



Javascript 程式碼很簡單：

```
function printCurrentPage() {  
    ...var pp = this.getPrintParams();  
    ...pp.firstPage = this.pageNum;  
    ...pp.lastPage = pp.firstPage;  
    ...this.print(pp);  
}
```

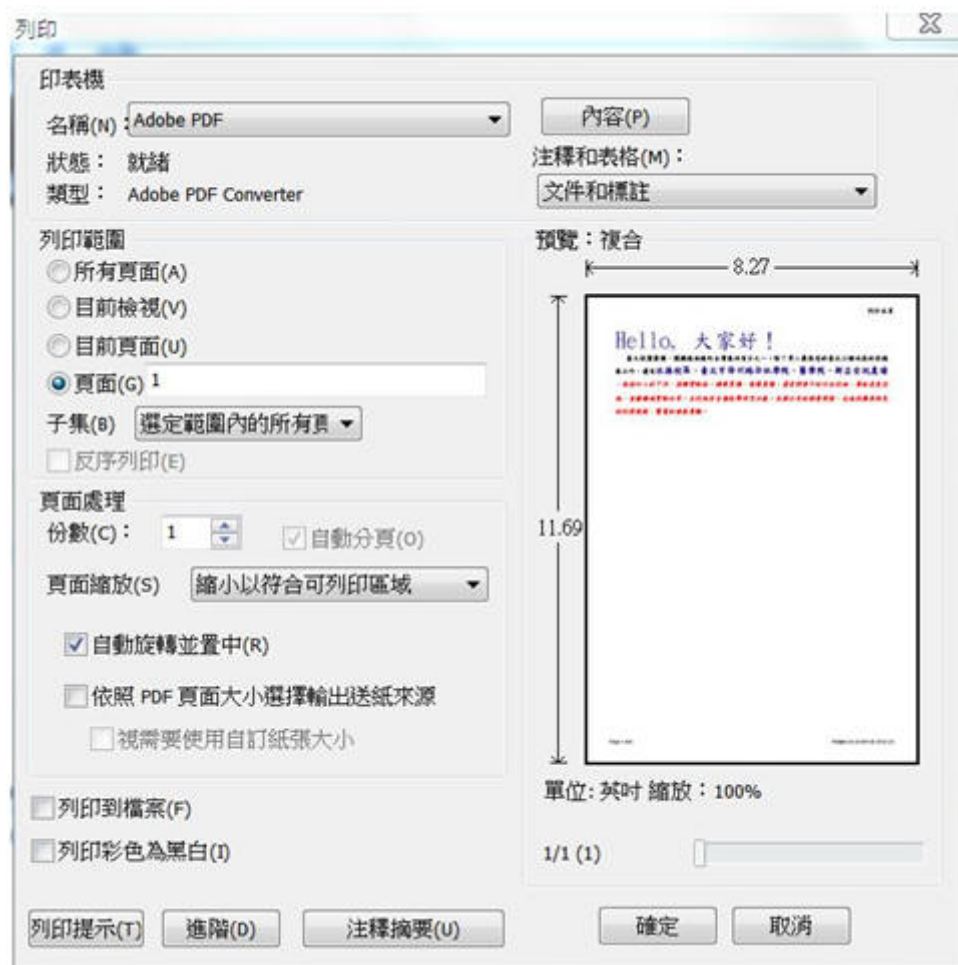
產生的結果如下圖：



右上角會出現<列印本頁>文字，當滑鼠移上去時會顯示可以click按下的指標，按下滑鼠左鍵後，會跳出alert 視窗：



接著按下確定，會出現列印對話方塊：



上述這個功能可以用在隱藏menu bar 及tool bar 時，使用者可以於頁面上直接做列印動作。

結語：

在尚未接觸用程式產生PDF 檔案之前，對PDF只有很粗淺的認識，猜測大致上應該就只是某種檔案格式。真正接觸後才發現真是挖到寶庫，尤其是有這麼強大函式庫可供使用，對程式開發者來說簡直是如虎添翼。還有沒有其他功能尚未介紹？答案是有的，而且還很多，再找時間介紹給各位讀者。各位若有需求，可以先從參考資料中開始找起，也許您會得到更多。

## 參考資料

### 1. 主要教學網站

<http://itextpdf.com/index.php> <http://www.mikesdotnetting.com/Article/80/Create-PDFs-in-ASP.NET-getting-started-with-iTextSharp>

### 2. 浮水印Watermark 製作<http://footheory.com/blogs/donnfelker/archive/2008/05/11/using-itextsharp-to-watermark-write-text-to-existing-pdf-s.aspx>

### 3. 書籤Bookmark製作

<http://www.mazsoft.com/blog/post/2008/04/30/Code-sample-for-using-iTextSharp-PDF-library.aspx>

### 4. Footer 製作

<http://stackoverflow.com/questions/1032614/itextsharp-creating-a-footer-page-of>

版權所有 © 國立台灣大學計算機及資訊網路中心 All Rights Reserved.

電話：02-33665022 或 3366-5023 傳真：02-23637204

讀者意見信箱：ntuccepaper@ntu.edu.tw

地址：10617 臺北市羅斯福路四段一號

建議最佳螢幕解析度 1024\*768