عنوان: **نحوه ذخیره شدن متن در فایل‌های PDF**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۹/۳۰ ۱۳:۳۸
آدرس: www.dotnettips.info
گروه‌ها: iTextSharp, PdfReport, PDF

تبدیل بی عیب و نقص یک فایل PDF (انواع و اقسام آن‌ها) به متن قابل درک بسیار مشکل است. در ادامه بررسی خواهیم کرد که چرا.
برخلاف تصور عموم، ساختار یک صفحه PDF شبیه به یک صفحه فایل Word نیست. این صفحات درحقیقت نوعی Canvas برای نقاشی هستند. در این بوم نقاشی، شکل، تصویر، متن و غیره در مختصات خاصی قرار خواهند گرفت. حتی کلمه «متن» می‌تواند به صورت سه حرف در سه مختصات خاص یک صفحه PDF نقاشی شود. برای درک بهتر این مورد نیاز است سورس یک صفحه PDF را بررسی کرد.

نحوه استخراج سورس یک صفحه PDF

```
using System.Diagnostics;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.pdf;

namespace TestReaders
{
    class Program
    {
        static void writePdf()
        {
            using (var document = new Document(PageSize.A4))
            {
                var writer = PdfWriter.GetInstance(document, new FileStream("test.pdf",
FileMode.Create));
                document.Open();

                document.Add(new Paragraph("Test"));

                PdfContentByte cb = writer.DirectContent;
                BaseFont bf = BaseFont.CreateFont();
                cb.BeginText();
                cb.SetFontAndSize(bf, 12);
                cb.MoveText(88.66f, 367);
                cb.ShowText("ld");
                cb.MoveText(-22f, 0);
                cb.ShowText("Wor");
                cb.MoveText(-15.33f, 0);
                cb.ShowText("llo");
                cb.MoveText(-15.33f, 0);
                cb.ShowText("He");
                cb.EndText();

                PdfTemplate tmp = cb.CreateTemplate(250, 25);
                tmp.BeginText();
                tmp.SetFontAndSize(bf, 12);
                tmp.MoveText(0, 7);
                tmp.ShowText("Hello People");
                tmp.EndText();
                cb.AddTemplate(tmp, 36, 343);
            }

            Process.Start("test.pdf");
        }

        private static void readPdf()
        {
            var reader = new PdfReader("test.pdf");
            int intPageNum = reader.NumberOfPages;
            for (int i = 1; i <= intPageNum; i++)
            {
                byte[] contentBytes = reader.GetPageContent(i);
                File.WriteAllBytes("page-" + i + ".txt", contentBytes);
            }
            reader.Close();
        }
```

```
        static void Main(string[] args)
        {
            writePdf();
            readPdf();
        }
    }
}
```

فایل PDF تولیدی حاوی سه عبارت کامل و مفهوم می‌باشد:

Test

Hello World
Hello People

اگر علاقمند باشید که سورس واقعی صفحات یک فایل PDF را مشاهده کنید، نحوه انجام آن توسط کتابخانه iTextSharp به صورت فوق است.
هرچند متد GetPageContent آرایه‌ای از بایت‌ها را بر می‌گرداند، اما اگر حاصل نهایی را در یک ادیتور متنی باز کنیم، قابل مطالعه و خواندن است. برای مثال، سورس مثال فوق (محتوای فایل page-1.txt تولید شده) به نحو زیر است:

```
q
BT
36 806 Td
0 -18 Td
/F1 12 Tf
(Test)Tj
0 0 Td
ET
Q
BT
/F1 12 Tf
```

```
88.66 367 Td
(ld)Tj
-22 0 Td
(Wor)Tj
-15.33 0 Td
(llo)Tj
-15.33 0 Td
(He)Tj
ET
q 1 0 0 1 36 343 cm /Xf1 Do Q
```

و تفسیر این عملگرها به این ترتیب است:

```
SaveGraphicsState(); // q
BeginText(); // BT
MoveTextPos(36, 806); // Td
MoveTextPos(0, -18); // Td
SelectFontAndSize("/F1", 12); // Tf
ShowText("(Test)"); // Tj
MoveTextPos(0, 0); // Td
EndTextObject(); // ET
RestoreGraphicsState(); // Q
BeginText(); // BT
SelectFontAndSize("/F1", 12); // Tf
MoveTextPos(88.66, 367); // Td
ShowText("(ld)"); // Tj
MoveTextPos(-22, 0); // Td
ShowText("(Wor)"); // Tj
MoveTextPos(-15.33, 0); // Td
ShowText("(llo)"); // Tj
MoveTextPos(-15.33, 0); // Td
ShowText("(He)"); // Tj
EndTextObject(); // ET
SaveGraphicsState(); // q
TransMatrix(1, 0, 0, 1, 36, 343); // cm
XObject("/Xf1"); // Do
RestoreGraphicsState(); // Q
```

همانطور که ملاحظه می‌کنید کلمه Test به مختصات خاصی انتقال داده شده و سپس به کمک اطلاعات فونت F1، ترسیم می‌شود. تا اینجا استخراج متن از فایل‌های PDF ساده به نظر می‌رسد. باید به دنبال Tj گشت و حروف مرتبط با آن‌را ذخیره کرد. اما در مورد «ترسیم» عبارات hello world و hello people اینطور نیست. عبارت hello world به حروف متفاوتی تقسیم شده و سپس در مختصات مشخصی ترسیم می‌گردد. عبارت hello people به صورت یک شیء ذخیره شده در قسمت منابع فایل PDF، بازیابی و نمایش داده می‌شود و اصلا در سورس صفحه جاری وجود ندارد.

این تازه قسمتی از نحوه عملکرد فایل‌های PDF است. در فایل‌های PDF می‌توان قلم‌ها را مدفون ساخت. همچنین این قلم‌ها نیز تنها زیر مجموعه‌ای از قلم اصلی مورد استفاده هستند. برای مثال اگر عبارت Test قرار است نمایش داده شود، فقط اطلاعات T، e و s در فایل نهایی PDF قرار می‌گیرند. به علاوه امکان تغییر کلی شماره Glyph متناظر با هر حرف نیز توسط PDF writer وجود دارد. به عبارتی الزامی نیست که مشخصات اصلی فونت حتما حفظ شود.

شاید بعضی از PDFهای فارسی را دیده باشید که پس از کپی متن آن‌ها در برنامه Adobe reader و سپس paste آن در جایی دیگر، متن حاصل قابل خواندن نیست. علت این است که نحوه ذخیره سازی قلم مورد استفاده کاملا تغییر کرده است و برای بازیابی متن اینگونه فایل‌ها، استفاده از OCR ساده‌ترین روش است. برای نمونه در این قلم جدید مدفون شده، دیگر شماره کاراکتر 0x41 مساوی A نیست. بنابر سلیقه PDF writer این شماره به Glyph دیگری انتساب داده شده و چون قلم و مشخصات هندسی Glyph مورد استفاده در فایل PDF ذخیره می‌شود، برای نمایش این نوع فایل‌ها هیچگونه مشکلی وجود ندارد. اما متن آن‌ها به سادگی قابل بازیابی نیست.