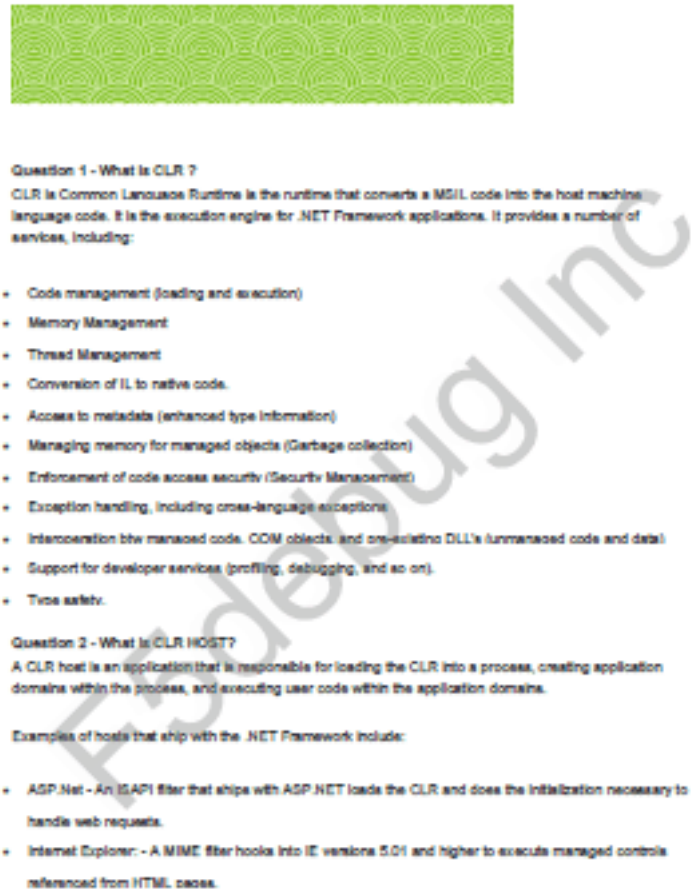
عنوان: نامرئی کردن Watermarkهای صفحات فایل‌های PDF توسط iTextSharp
نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۳۰ ۸:۰
آدرس: www.dotnettips.info
گروه‌ها: iTextSharp, Watermark

احتمالا بارها با PDFهایی که یک Watermark بزرگ را در میانه صفحات خود دارند، برخورد داشته‌اید و متاسفانه در اغلب اوقات استفاده ناصحیحی از این قابلیت صورت می‌گیرد. هدف از Watermark دار کردن صفحات PDF، ذکر جملاتی مانند «آزمایشی بودن» یا «محرمانه بودن» است که در هر دو حالت نباید به صورت عمومی منتشر شوند. اما اگر قرار است مطلبی را به صورت عمومی منتشر کنیم، این روش، بدترین حالت تبلیغی برای یک شخص یا شرکت خواهد بود؛ چون مانع خواندن روان متن شده و اعصاب مصرف کننده را به هم خواهد ریخت. بنابراین هیچگونه جنبه تبلیغی مثبتی را در نهایت برای تهیه کننده به همراه نخواهد داشت.
برای نمونه فایل نمونه سؤالات مصاحبات دات را از اینجا دریافت کنید . یک چنین شکلی دارد:



Question 1 - What is CLR ?
CLR is Common Language Runtime is the runtime that converts a MSIL code into the host machine language code. It is the execution engine for .NET Framework applications. It provides a number of services, including:

- Code management (loading and execution)
- Memory Management
- Thread Management
- Conversion of IL to native code.
- Access to metadata (enhanced type information)
- Managing memory for managed objects (Garbage collection)
- Enforcement of code access security (Security Management)
- Exception handling, including cross-language exceptions
- Interoperation btw managed code, COM objects, and pre-existing DLL's (unmanaged code and data)
- Support for developer services (profiling, debugging, and so on).
- Type safety.

Question 2 - What is CLR HOST?
A CLR host is an application that is responsible for loading the CLR into a process, creating application domains within the process, and executing user code within the application domains.

Examples of hosts that ship with the .NET Framework include:

- ASP.Net - An ISAPI filter that ships with ASP.NET loads the CLR and does the initialization necessary to handle web requests.
- Internet Explorer: - A MIME filter hooks into IE versions 5.01 and higher to execute managed controls referenced from HTML pages.

خوب! چطور می‌توان این Watermark را حذف یا حداقل نامرئی کرد؟
برای پاسخ به این سؤال نیاز است ابتدا با نحوه Watermark دار کردن صفحات یک فایل PDF آشناشویم.

الف) ایجاد یک فایل PDF ساده

```
private static void createPdfFile(string pdfFile)
        {
            using (var fs = new FileStream(pdfFile, FileMode.Create, FileAccess.Write, FileShare.None))
            {
                using (var doc = new Document(PageSize.A4))
                {
                    using (var witier = PdfWriter.GetInstance(doc, fs))
                    {
```

```
                        doc.Open();
                        for (int i = 1; i <= 5; i++)
                        {
                            doc.NewPage();
                            doc.Add(new Paragraph(String.Format("This is page {0}", i)));
                        }
                        doc.Close();
                    }
                }
            }
        }
```

در اینجا یک فایل PDF با 5 صفحه ایجاد می‌شود.

ب) افزودن Watermark به فایل PDF تهیه شده

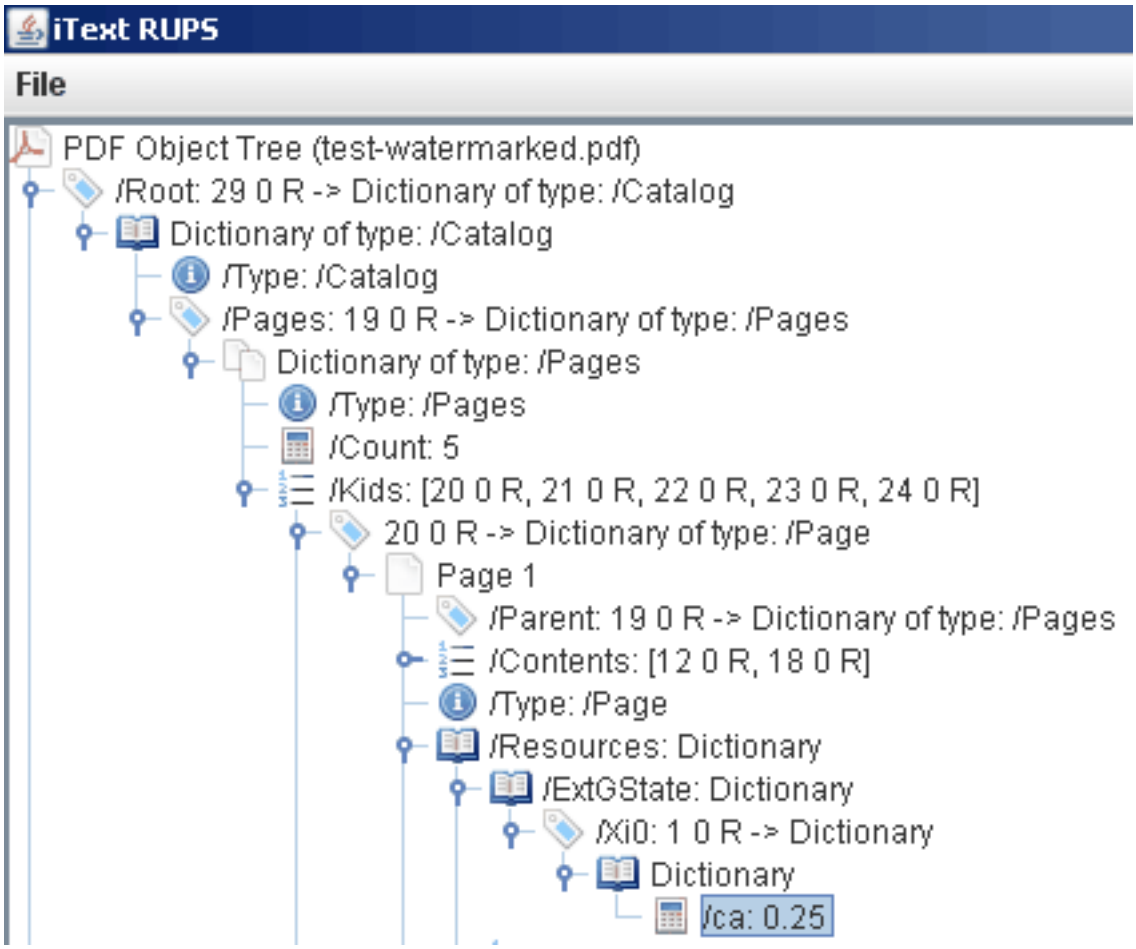
```
private static void addWatermark(string pdfFile, string watermarkedFile, string watermarkText)
        {
            FontFactory.Register("c:\\windows\\fonts\\tahoma.ttf");
            var tahoma = FontFactory.GetFont("Tahoma", BaseFont.IDENTITY_H, 40, Font.NORMAL,
BaseColor.BLACK);

            var reader = new PdfReader(pdfFile);
            using (var fileStream = new FileStream(watermarkedFile, FileMode.Create, FileAccess.Write,
FileShare.None))
            {
                using (var stamper = new PdfStamper(reader, fileStream))
                {
                    int pageCount = reader.NumberOfPages;
                    for (int i = 1; i <= pageCount; i++)
                    {
                        var rect = reader.GetPageSize(i);
                        var cb = stamper.GetUnderContent(i);
                        var gState = new PdfGState();
                        gState.FillOpacity = 0.25f;
                        cb.SetGState(gState);

                        ColumnText.ShowTextAligned(
                            canvas: cb,
                            alignment: Element.ALIGN_CENTER,
                            phrase: new Phrase(watermarkText, tahoma),
                            x: rect.Width / 2,
                            y: rect.Height / 2,
                            rotation: 45f,
                            runDirection: PdfWriter.RUN_DIRECTION_LTR,
                            arabicOptions: 0);
                    }
                }
            }
        }
```

در متد فوق pdfFile نام و مسیر فایل PDF ایی است که قرار است به صفحات آن Watermark اضافه شود. نام و مسیر فایل خروجی توسط watermarkedFile مشخص می‌شود و watermarkText متنی است که در میانه صفحه نمایش داده خواهد شد. در اینجا توسط PdfReader، فایل موجود گشوده می‌شود. به این ترتیب می‌توان به تک تک صفحات این فایل دسترسی یافت. از PdfStamper برای نوشتن در این فایل باز شده استفاده می‌کنیم. متد stamper.GetUnderContent، لایه زیرین متن صفحات را در اختیار ما قرار می‌دهد. اگر علاقمند به نوشتن بر روی لایه رویی متون هستید از متد stamper.GetOverContent استفاده کنید. در اینجا از PdfGState برای مشخص سازی میزان شفافیت متن درحال نمایش، با مقدار دهی FillOpacity آن استفاده شده است. نهایتا از متد ColumnText.ShowTextAligned برای نمایش متن مورد نظر در مکان و زاویه‌ای مشخص استفاده می‌کنیم. این متد با زبان فارسی سازگاری دارد و run direction آن قابل تنظیم است.

ج) آشنایی با ساختار سطح پایین Watermark اضافه شده به صفحات

تقریبا اکثر Watermarkهایی که بر روی صفحات PDF درج می‌شوند، نیمه شفاف هستند تا بتوان متن زیر آن‌ها را مطالعه کرد. این شفافیت همانطور که ذکر شد توسط مقدار دهی شیء PdfGState حاصل می‌شود. اگر فایل PDF تولیدی در قسمت ب را توسط

برنامه iText Rups باز کنیم، به شکل زیر خواهیم رسید:



هدف ما این است که به شیء PdfGState موجود در هر صفحه، دسترسی یافته و مقدار FillOpacity آن‌را صفر کنیم. به این ترتیب این Watermark، کاملا شفاف یا نامرئی خواهد شد. در PDF نهایی، چیزی به نام شیء PdfGState وجود ندارد، بلکه با یک سری Dictionary و Array سر و کار داریم.

همانطور که در شکل فوق ملاحظه می‌کنید، برای رسیدن به Gstateها باید مراحل زیر طی شوند:

1- فایل PDF گشوده شده و سپس به هر صفحه دسترسی یافت.

2- نیاز است RESOURCES صفحه جاری استخراج شوند.

3- در این منابع، باید EXTGSTATE را که همان PdfGStateها هستند، بیابیم.

4- سپس مقدار ca این EXTGSTATE یافت شده را به صفر مقدار دهی کنیم.

```
private static void removeWatermark(string watermarkedFile, string unwatermarkedFile)
        {
            PdfReader.unethicalreading = true;
            PdfReader reader = new PdfReader(watermarkedFile);
            reader.RemoveUnusedObjects();
            int pageCount = reader.NumberOfPages;
            for (int i = 1; i <= pageCount; i++)
            {
                var page = reader.GetPageN(i);
                PdfDictionary resources = page.GetAsDict(PdfName.RESOURCES);
                PdfDictionary extGStates = resources.GetAsDict(PdfName.EXTGSTATE);
                if (extGStates == null)
                    continue;

                foreach (PdfName name in extGStates.Keys)
```

```
                {
                    var obj = extGStates.Get(name);
                    PdfDictionary extGStateObject = (PdfDictionary)PdfReader.GetPdfObject(obj);
                    var stateNumber = extGStateObject.Get(PdfName.ca_);
                    if (stateNumber == null)
                        continue;

                    var caNumber = (PdfNumber)PdfReader.GetPdfObject(stateNumber);
                    if (caNumber.FloatValue != 1f)
                    {
                        extGStateObject.Remove(PdfName.ca_);
                        //با تنظیم مقدار به صفر، نامرئی خواهد شد
                        extGStateObject.Put(PdfName.ca_, new PdfNumber(0f));
                    }
                }
            }

            using (FileStream fs = new FileStream(unwatermarkedFile, FileMode.Create, FileAccess.Write,
FileShare.None))
            {
                using (PdfStamper stamper = new PdfStamper(reader, fs))
                {
                    stamper.SetFullCompression();
                    stamper.Close();
                }
            }
        }
```

نحوه پیاده سازی مراحل یاد شده را در کدهای فوق ملاحظه می‌کنید. ابتدا توسط PdfReader فایل موجود باز شده و سپس تک تک صفحات آن‌را بررسی می‌کنیم. در این صفحات اگر EXTGSTATE ایی یافت شد، مقدار ca آن‌را به صفر تنظیم خواهیم کرد. از مقدار 1 صرفنظر شده، چون این مقدار عموما برای Watermark دار کردن صفحات استفاده نمی‌شود.
در این متد، watermarkedFile فایلی است که باید watermark آن نامرئی شود و unwatermarkedFile فایل تولیدی حاصل است.

Question 1 - What is CLR ?

CLR is Common Language Runtime is the runtime that converts a MSIL code into the host machine language code. It is the execution engine for .NET Framework applications. It provides a number of services, including:

- Code management (loading and execution)
- Memory Management
- Thread Management
- Conversion of IL to native code.
- Access to metadata (enhanced type information)
- Managing memory for managed objects (Garbage collection)
- Enforcement of code access security (Security Management)
- Exception handling, including cross-language exceptions
- Interoperation btw managed code, COM objects, and pre-existing DLL's (unmanaged code and data)
- Support for developer services (profiling, debugging, and so on).
- Type safety.

Question 2 - What is CLR HOST?

A CLR host is an application that is responsible for loading the CLR into a process, creating application domains within the process, and executing user code within the application domains.

Examples of hosts that ship with the .NET Framework include:

- ASP.Net - An ISAPI filter that ships with ASP.NET loads the CLR and does the initialization necessary to handle web requests.
- Internet Explorer: - A MIME filter hooks into IE versions 5.01 and higher to execute managed controls referenced from HTML pages.