عنوان: **نمایش یک فایل PDF در WPF ، WinForms و سیلورلایت**
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۴/۲۰ ۲۲:۵۴:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: WPF

شاید PDF را بشود تنها فرمت گزارشگیری دانست که همه‌جا و در تمام سیستم عامل‌ها پشتیبانی می‌شود. از ویندوز تا لینوکس از وب تا WPF تا سیلورلایت تا همه جا و از همه مهم‌تر اینکه خروجی آن دقیقا همان چیزی است که کاربر نهایی می‌خواهد: من می‌خوام اون چیزی رو که می‌بینم، دقیقا همان را، بدون کم و کاست و با همان صفحه بندی، بتوانم چاپ کنم.
برای تولید PDF می‌شود از کتابخانه‌ی iTextSharp استفاده کرد اما برای نمایش آن حداقل در ویندوز بهترین راه حل استفاده از COM Components شرکت Adobe است که به همراه برنامه رایگان Adobe PDF reader ارائه می‌شود. در ادامه نحوه‌ی استفاده از این Active-X را بررسی خواهیم کرد.

نمایش PDF در WPF
در تمام حالت‌ها هدف این است که به نحوی به اکتیوایکس شرکت Adobe دسترسی پیدا کنیم؛ یا با اضافه کردن آن به پروژه یا استفاده از امکانات یکپارچه مرورگرها. در WPF از زمان ارائه سرویس پک یک دات نت سه و نیم (به بعد)، کنترل مرورگر وب هم به جمع کنترل‌های قابل استفاده در آن اضافه شده است. در اینجا به سادگی چند سطر زیر می‌شود یک فایل PDF را در WPF نمایش داد:

```
<Window x:Class="WpfAppTests.MainWindow"
       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
       Title="Pdf Report" Height="495" WindowState="Maximized"
       WindowStartupLocation="CenterScreen" Width="703">
   <Grid>
       <WebBrowser x:Name="WebBrowser1"/>
   </Grid>
</Window>
```

و بعد هم در کدهای برنامه تنها کافی است که مقدار Source کنترل WebBrowser را مقدار دهی کرد:

```
WebBrowser1.Source = new Uri(PdfFilePath);
```

نمایش PDF در WinForms
اکتیوایکس نمایش دهنده PDF شرکت Adobe اساسا در فایل ذیل قرار گرفته است:

```
C:\Program Files\Common Files\Adobe\Acrobat\ActiveX\AcroPDF.dll
```

بنابراین برای استفاده از آن در یک برنامه‌ی WinForms باید مراحل ذیل طی شود:
الف) در VS.NET از طریق منوی Tools گزینه‌ی Choose toolbox items ، برگه‌ی Com components را انتخاب کنید.
ب) سپس گزینه‌ی Adobe PDF reader که به همان مسیر dll فوق اشاره می‌کند را انتخاب نمائید و بر روی دکمه‌ی OK کلیک کنید.
ج) اکنون این کنترل جدید را بر روی فرم برنامه قرار دهید. به صورت خودکار COMReference های متناظر به پروژه اضافه می‌شوند.

اکنون نحوه‌ی استفاده از این شیء COM به همراه آزاد سازی منابع مرتبط به شرح زیر خواهند بود:

```
using System.Windows.Forms;

namespace WindowsFormsAppTests
```

```
{
   public partial class Form1 : Form
   {
       public Form1()
       {
           InitializeComponent();
           this.Load += Form1_Load;
           this.FormClosing += Form1_FormClosing;
       }

       void Form1_FormClosing(object sender, FormClosingEventArgs e)
       {
           axAcroPDF1.Dispose();
       }

       void Form1_Load(object sender, System.EventArgs e)
       {
           axAcroPDF1.LoadFile(PdfFilePath);
           axAcroPDF1.setShowToolbar(true);
           axAcroPDF1.Show();
       }
   }
}
```

نمایش PDF در Silverlight

در Silverlight هم از نسخه‌ی 4 به بعد کنترل WebBrowser همانند آنچه که در WPF موجود است، اضافه شده است؛ اما این کنترل فقط در حالت اجرای در خارج از مرورگر برنامه Silverlight در دسترس می‌باشد. بنابراین روش دیگری را باید انتخاب کرد. این روش بر اساس تعامل سیلورلایت با کدهای HTML صفحه کار می‌کند. یک IFrame مخفی را در صفحه بالای شیء مرتبط با سیلورلایت قرار خواهیم داد. سپس در سیلورلایت Src این IFrame را به مسیر فایل PDF تنظیم می‌کنیم و همین. به این ترتیب فایل PDF نمایش داده می‌شود.

این IFrame به صورت زیر در همان صفحه‌ی aspx ایی که object مرتبط با Silverlight نمایش داده می‌شود قرار می‌گیرد:

```
<iframe id="pdfFrame" style="visibility:hidden; position:absolute"><b>No Content</b></iframe>
<div id="silverlightControlHost">
```

سپس در کدهای سیلورلایت، ابتدا این IFrame یافت شده:

```
var iFrame = HtmlPage.Document.GetElementById("pdfFrame");
```

در ادامه بر اساس اطلاعات مکانی یک Grid ساده به نام pdfHost که در صفحه قرار گرفته، این iFrame بالاتر از سطح Grid (بر اساس z-index تنظیم شده) نمایش داده می‌شود:

```
var gt = pdfHost.TransformToVisual(Application.Current.RootVisual);
var offset = gt.Transform(new Point(0, 0));
var controlLeft = (int)offset.X;
var controlTop = (int)offset.Y;
iFrame.SetStyleAttribute("left", string.Format("{0}px", controlLeft));
iFrame.SetStyleAttribute("top", string.Format("{0}px", controlTop));
iFrame.SetStyleAttribute("visibility", "visible");
iFrame.SetStyleAttribute("height", string.Format("{0}px", pdfHost.ActualHeight));
iFrame.SetStyleAttribute("width", string.Format("{0}px", pdfHost.ActualWidth));
iFrame.SetStyleAttribute("z-index", "1000");
```

و در آخر نام فایلی را که می‌خواهیم مشاهده کنیم به یک صفحه‌ی aspx در همان سایت ارسال می‌کنیم:

```
iFrame.SetProperty("src", "ShowPdf.aspx?file=" + fileName);
```

کدهای این صفحه در حد یک Response.Redirect ساده برای نمایش دادن فایل pdf در مرورگر کافی هستند. در کل در اینجا

سیلورلایت تنها نقش انتخاب فایل را به عهده دارد و کار اصلی را خود مرورگر انجام می‌دهد.

نظرات خوانندگان

نویسنده: Yari AliReza 2010
تاریخ: ۱۳۹۰/۰۴/۲۱ ۰۹:۳۲:۲۲

خیلی ممنون که تجربیات خود را در اختیار بقیه می گذارید.

نویسنده: Naeim Rezaeian
تاریخ: ۱۳۹۰/۰۶/۱۲ ۰۰:۲۷:۰۲

دست شما درد نکنه فقط میشه بگید که چطوری میشه همه تولبار و اون منوی که روی فایل میاد رو غیر فعال کرد .با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۶/۱۲ ۰۸:۲۵:۴۵

در حالت windows forms ، فقط کافی است بنویسید: axAcroPDF1.setShowToolbar(false)
در دو حالت دیگر که از مرورگر استفاده می‌شود، اینکار بی‌فایده است چون فایل مورد نظر از کش مرورگر قابل استخراج است.
اما اگر نیاز به حداقل کنترل وجود داشت باید تگ object را به صورت زیر درست کرد (کمی باید html نویسی کرد):

```
object type="application/pdf" data="file1.pdf#navpanes=0&scrollbar=0 &toolbar=0" width="500"
"height="650
```