

برنامه رایگان Adobe reader یک سری خط فرمان دارد که توسط آن‌ها می‌توان فایل‌های PDF را مستقیماً به چاپگر ارسال کرد. در ادامه قطعه کدی را ملاحظه خواهید کرد که انجام اینکار را کپسوله می‌کند:

```
using System;
using System.Diagnostics;
using System.IO;
using System.Management;
using Microsoft.Win32;

namespace PdfFilePrinter
{
    /// <summary>
    /// Executes the Adobe Reader and prints a file while suppressing the Acrobat print
    /// dialog box, then terminating the Reader.
    /// </summary>
    public class AcroPrint
    {
        /// <summary>
        /// The Adobe Reader or Adobe Acrobat path such as 'C:\Program Files\Adobe\Adobe Reader
        X\AcroRd32.exe'.
        /// If it's not specified, the InstalledAdobeReaderPath property value will be used.
        /// </summary>
        public string AdobeReaderPath { set; get; }

        /// <summary>
        /// Returns the default printer name.
        /// </summary>
        public string DefaultPrinterName
        {
            get
            {
                var query = new ObjectQuery("SELECT * FROM Win32_Printer");
                using (var searcher = new ManagementObjectSearcher(query))
                {
                    foreach (var mo in searcher.Get())
                    {
                        if (((bool?)mo["Default"]) ?? false)
                            return mo["Name"] as string;
                    }
                }
                return string.Empty;
            }
        }

        /// <summary>
        /// The name and path of the PDF file to print.
        /// </summary>
        public string PdfFilePath { set; get; }

        /// <summary>
        /// Name of the printer such as '\\PrintServer\HP LaserJet'.
        /// If it's not specified, the DefaultPrinterName property value will be used.
        /// </summary>
        public string PrinterName { set; get; }

        /// <summary>
        /// Returns the HKEY_CLASSES_ROOT\Software\Adobe\Acrobat\Exe value.
        /// If AcroRd32.exe does not exist, returns string.Empty
        /// </summary>
        public string InstalledAdobeReaderPath
        {
            get
            {
                var acroRd32Exe = Registry.ClassesRoot.OpenSubKey(@"Software\Adobe\Acrobat\Exe",
                writable: false);
                if (acroRd32Exe == null)
                    return string.Empty;

                var exePath = acroRd32Exe.GetValue(string.Empty) as string;
                if (string.IsNullOrEmpty(exePath))
                    return string.Empty;
            }
        }
    }
}
```

```

        exePath = exePath.Trim(new[] { '"' });
        return File.Exists(exePath) ? exePath : string.Empty;
    }
}

/// <summary>
/// Executes the Adobe Reader and prints a file while suppressing the Acrobat print
/// dialog box, then terminating the Reader.
/// </summary>
/// <param name="timeout">The amount of time, in milliseconds, to wait for the associated
process to exit. The maximum is the largest possible value of a 32-bit integer, which represents
infinity to the operating system.</param>
public void PrintPdfFile(int timeout = Int32.MaxValue)
{
    if (!File.Exists(PdfFilePath))
        throw new ArgumentException(PdfFilePath + " does not exist.");

    var args = string.Format("/N /T \"{0}\" \"{1}\"", PdfFilePath, getPrinterName());
    var process = startAdobeProcess(args);
    if (!process.WaitForExit(timeout))
        process.Kill();
}

private Process startAdobeProcess(string arguments = "")
{
    var startInfo = new ProcessStartInfo
    {
        FileName = this.getExePath(),
        Arguments = arguments,
        CreateNoWindow = true,
        ErrorDialog = false,
        UseShellExecute = false,
        Verb = "print"
    };

    return Process.Start(startInfo);
}

private string getPrinterName()
{
    var printer = PrinterName;
    if (string.IsNullOrEmpty(printer))
        printer = DefaultPrinterName;

    if (string.IsNullOrEmpty(printer))
        throw new ArgumentException("Please set the PrinterName.");

    return printer;
}

private string getExePath()
{
    var exePath = AdobeReaderPath;
    if (string.IsNullOrEmpty(exePath) || !File.Exists(exePath))
        exePath = InstalledAdobeReaderPath;

    if (string.IsNullOrEmpty(exePath))
        throw new ArgumentException("Please set the full path of the AcroRd32.exe or
Acrobat.exe.");

    return exePath;
}
}
}

```

توضیحات:

استفاده ابتدایی از کلاس فوق به نحو زیر است:

```

new AcroPrint
{
    PdfFilePath = @"D:\path\test.pdf"
}.PrintPdfFile();

```

به این ترتیب فایل PDF ذکر شده به چاپگر پیش فرض سیستم ارسال می‌شود.

ملاحظات:

- کدهای فوق نیاز به ارجاعی به اسمبلی استاندارد System.Management.dll نیز دارند.
- اگر علاقمند بودید که چاپگر خاصی را معرفی کنید (برای مثال یک چاپگر تعریف شده در شبکه)، می‌توانید خاصیت PrinterName را مقدار دهی نمائید.
- محل نصب Adobe reader از رجیستری ویندوز استخراج می‌شود. اما اگر محل نصب برنامه استاندارد نبود، نیاز است خاصیت AdobeReaderPath مقدار دهی گردد.
- تحت هر شرایطی برنامه Adobe reader ظاهر خواهد شد؛ حتی اگر در حین آغاز پروسه سعی در مخفی کردن پنجره آن نمائید.
- اینکار به عمد جهت مسایل امنیتی در این برنامه در نظر گرفته شده است تا کاربر بداند که پروسه چاپ آغاز شده است.

نظرات خوانندگان

نویسنده: ایمان محمدی
تاریخ: ۰۲۴ ۱۳۹۱/۱۱/۲۴

فکر کنم در فایل‌های pdf قابلیت وجود داره که دستور پرینت رو می‌تونه ذخیره کنه تا بهنگام باز شدن فایل دستور پرینت (یا پرینت دیالوگ) اجرا بشه. کاربرد این حالت برای وب ممکنه مناسب باشه ، شاید بد نیست چنین قابلیتی در پروژه PdfReport داشته باشید.

نویسنده: وحید نصیری
تاریخ: ۰۳۷ ۱۳۹۱/۱۱/۲۴

وجود داره. این تنظیم رو اضافه کنید:

```
return new PdfReport().DocumentPreferences(doc =>
{
    //...
    doc.PrintingPreferences(new PrintingPreferences
    {
        ShowPrintDialogAutomatically = true
    });
})
```

نویسنده: ایمان محمدی
تاریخ: ۱:۱۷ ۱۳۹۱/۱۱/۲۴

چه خوب ، نکته جالب اینه pdf خوان داخلی کروم هم از این قابلیت پشتیبانی میکنه و برای یک برنامه وب می‌تونه تداعی کننده print perview یک برنامه ویندوزی باشه.

نویسنده: علی پناهی
تاریخ: ۱۳:۴۲ ۱۳۹۳/۰۱/۲۰

با مقدار Timeout مشکل دارم، مثلا مقدار 10 که میدم در یک سیستم قدیمی چاپ انجام نمیشه و برای اطمینان از چاپ 50 که میدم ، چون دو برگه برای چاپ دارم یکی برای مشتری یکی برای مغازه دار دو دقیقه طول میکشه.

نویسنده: محسن خان
تاریخ: ۱۴:۴۴ ۱۳۹۳/۰۱/۲۰

مقدار پیش فرضش که در متد مشخص شده، برای تمام سیستم‌ها کافی هست.

نویسنده: علی پناهی
تاریخ: ۷:۱۱ ۱۳۹۳/۰۱/۲۱

اصلاحیه اعداد من 10000 و 50000 می‌باشد.
من زمان تعیین کرده بودم که کاربر پنجره را نبندد و خودش بسته بشود،
پس باید به کاربر بگویم خودش پنجره Adobe Reader را ببندد.
اگر این مورد را بشود به صورت دیگه ای حل کرد، بهتر و کاربردی تره.