

با استفاده از اشیاء Com همراه با Acrobat SDK می‌توان تمام صفحات یک فایل PDF را تبدیل به تصویر کرد. این SDK به همراه نگارش کامل Adobe Acrobat نیز بر روی سیستم نصب می‌شود و یا می‌توان آن را به صورت جداگانه از سایت Adobe دریافت کرد.

<http://www.adobe.com/devnet/acrobat/downloads.html>

پس از آن، برای تبدیل صفحات یک فایل PDF به تصویر، مراحل زیر باید طی شود:

الف) وهله سازی از شیء AcroExch.PDDoc

در صورتیکه SDK یاد شده بر روی سیستم نصب نباشد، این وهله سازی با شکست مواجه خواهد شد و همچنین باید دقت داشت که این SDK به همراه نگارش رایگان Adobe reader ارائه نمی‌شود.

ب) گشودن فایل PDF به کمک شیء Com وهله سازی شده (pdfDoc.Open)

ج) دریافت اطلاعات صفحه مورد نظر (pdfDoc.AcquirePage)

د) کپی این اطلاعات به درون clipboard ویندوز (pdfPage.CopyToClipboard)

به این ترتیب به یک تصویر Bmp قرار گرفته شده در clipboard ویندوز خواهیم رسید

ه) مرحله بعد تغییر ابعاد و ذخیره سازی این تصویر نهایی است.

کدهای زیر، روش انجام این مراحل را بیان می‌کنند:

```
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.IO;
using System.Runtime.InteropServices;
using System.Threading;
using System.Windows.Forms;
using Acrobat; //Add a Com Object ref. to "Adobe Acrobat 10.0 Type Library" => Program
Files\Adobe\Acrobat 10.0\Acrobat\acrobat.tlb
using Microsoft.Win32;

namespace PdfThumbnail.Lib
{
    public static class PdfToImage
    {
        const string AdobeObjectsErrorMessage = "Failed to create the PDF object.";
        const string BadFileErrorMessage = "Failed to open the PDF file.";
        const string ClipboardError = "Failed to get the image from clipboard.";
        const string SdkError = "This operation needs the Acrobat
        SDK(http://www.adobe.com/devnet/acrobat/downloads.html), which is combined with the full version of
        Adobe Acrobat.";

        public static byte[] PdfPageToPng(string pdfFilePath, int thumbWidth = 600, int thumbHeight =
        750, int pageNumber = 0)
        {
            byte[] imageData = null;
            runJob((pdfDoc, pdfRect) =>
            {
                imageData = pdfPageToPng(thumbWidth, thumbHeight, pageNumber, pdfDoc, pdfRect);
            }, pdfFilePath);
            return imageData;
        }

        public static void AllPdfPagesToPng(Action<byte[], int, int> dataCallback, string pdfFilePath,
        int thumbWidth = 600, int thumbHeight = 750)
        {
            runJob((pdfDoc, pdfRect) =>
            {
                var numPages = pdfDoc.GetNumPages();
                for (var pageNumber = 0; pageNumber < numPages; pageNumber++)
                {
                    var imageData = pdfPageToPng(thumbWidth, thumbHeight, pageNumber, pdfDoc,
                    pdfRect);
                    dataCallback(imageData, pageNumber + 1, numPages);
                }
            }, pdfFilePath);
        }
    }
}
```

```

    }

    static void runJob(Action<CAcroPDDoc, CAcroRect> job, string pdfFilePath)
    {
        if (!File.Exists(pdfFilePath))
            throw new InvalidOperationException(BadFileErrorMessage);

        var acrobatPdfDocType = Type.GetTypeFromProgID("AcroExch.PDDoc");
        if (acrobatPdfDocType == null || !isAdobeSdkInstalled)
            throw new InvalidOperationException(SdkError);

        var pdfDoc = (CAcroPDDoc)Activator.CreateInstance(acrobatPdfDocType);
        if (pdfDoc == null)
            throw new InvalidOperationException(AdobeObjectsErrorMessage);

        var acrobatPdfRectType = Type.GetTypeFromProgID("AcroExch.Rect");
        var pdfRect = (CAcroRect)Activator.CreateInstance(acrobatPdfRectType);

        var result = pdfDoc.Open(pdfFilePath);
        if (!result)
            throw new InvalidOperationException(BadFileErrorMessage);

        job(pdfDoc, pdfRect);

        releaseComObjects(pdfDoc, pdfRect);
    }

    public static byte[] ResizeImage(this Image image, int thumbWidth, int thumbHeight)
    {
        var srcWidth = image.Width;
        var srcHeight = image.Height;
        using (var bmp = new Bitmap(thumbWidth, thumbHeight, PixelFormat.Format32bppArgb))
        {
            using (var gr = Graphics.FromImage(bmp))
            {
                gr.SmoothingMode = SmoothingMode.HighQuality;
                gr.PixelOffsetMode = PixelOffsetMode.HighQuality;
                gr.CompositingQuality = CompositingQuality.HighQuality;
                gr.InterpolationMode = InterpolationMode.High;

                var rectDestination = new Rectangle(0, 0, thumbWidth, thumbHeight);
                gr.DrawImage(image, rectDestination, 0, 0, srcWidth, srcHeight,
GraphicsUnit.Pixel);

                using (var memStream = new MemoryStream())
                {
                    bmp.Save(memStream, ImageFormat.Png);
                    return memStream.ToArray();
                }
            }
        }
    }

    static bool isAdobeSdkInstalled
    {
        get
        {
            return Registry.ClassesRoot.OpenSubKey("AcroExch.PDDoc", writable: false) != null;
        }
    }

    private static Bitmap pdfPageToBitmap(int pageNumber, CAcroPDDoc pdfDoc, CAcroRect pdfRect)
    {
        var pdfPage = (CAcroPDPage)pdfDoc.AcquirePage(pageNumber);
        if (pdfPage == null)
            throw new InvalidOperationException(BadFileErrorMessage);

        var pdfPoint = (CAcroPoint)pdfPage.GetSize();

        pdfRect.Left = 0;
        pdfRect.Right = pdfPoint.x;
        pdfRect.Top = 0;
        pdfRect.Bottom = pdfPoint.y;

        pdfPage.CopyToClipboard(pdfRect, 0, 0, 100);

        Bitmap pdfBitmap = null;
        var thread = new Thread(() =>
        {
            var data = Clipboard.GetDataObject();
            if (data != null && data.GetDataPresent(DataFormats.Bitmap))

```

```

        pdfBitmap = (Bitmap)data.GetData(DataFormats.Bitmap);
    });
    thread.SetApartmentState(ApartmentState.STA);
    thread.Start();
    thread.Join();

    Marshal.ReleaseComObject(pdfPage);

    return pdfBitmap;
}

private static byte[] pdfPageToPng(int thumbWidth, int thumbHeight, int pageNumber, CAcroPDDoc pdfDoc, CAcroRect pdfRect)
{
    var pdfBitmap = pdfPageToBitmap(pageNumber, pdfDoc, pdfRect);
    if (pdfBitmap == null)
        throw new InvalidOperationException(ClipboardError);

    var pdfImage = pdfBitmap.GetThumbnailImage(thumbWidth, thumbHeight, null, IntPtr.Zero);
    // (+ 7 for template border)
    var imageData = pdfImage.ResizeImage(thumbWidth + 7, thumbHeight + 7);
    return imageData;
}

private static void releaseComObjects(CAcroPDDoc pdfDoc, CAcroRect pdfRect)
{
    pdfDoc.Close();
    Marshal.ReleaseComObject(pdfRect);
    Marshal.ReleaseComObject(pdfDoc);
}
}
}

```

و برای استفاده از آن خواهیم داشت:

```

using System;
using System.IO;
using System.Windows.Forms;
using PdfThumbnail.Lib;

namespace PdfThumbnail
{
    class Program
    {
        static void Main(string[] args)
        {
            var pdfPath = Application.StartupPath + @"\test.pdf";
            PdfToImage.AllPdfPagesToPng((pageImageData, pageNumber, numPages) =>
            {
                Console.WriteLine("Page {0}/{1}", pageNumber, numPages);
                File.WriteAllBytes(string.Format("{0}\\page-{1}.png", Application.StartupPath,
                pageNumber), pageImageData);
            }, pdfPath);
        }
    }
}

```

کدهای این قسمت را از اینجا نیز می‌توانید دریافت کنید:

[PdfThumbnail.zip](#)

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۸/۲۰ ۰:۲۰

امکان دریافت SDK فوق با IP ایرانی نیست. آنرا [از اینجا](#) هم می‌توانید دریافت کنید.

نویسنده: molana11
تاریخ: ۱۳۹۲/۰۱/۲۲ ۱۴:۴۷

با سلام.
می‌توان از این مثال در وب نیز استفاده کرد؟
باتشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۲۲ ۱۵:۳۰

- اگر سرور دار خودتون هستید و می‌تونید روی سرور وابستگی‌های مربوط به نگارش کامل Adobe Acrobat را نصب کنید و همچنین برنامه در حالت Full trust اجرا می‌شود؛ بله.
- راه حل‌های دیگری هم هستند که در قسمت اشتراک‌های سایت [مطرح شدند](#) .

نویسنده: molana11
تاریخ: ۱۳۹۲/۰۱/۲۲ ۱۵:۱۰

مهندس جان.
کیفیت تصویر تولیدی را چگونه می‌توان بالا برد؟
باتشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۱/۲۲ ۱۶:۲۹

- کیفیت بالایی داره.
- حداکثر از ResizeImage استفاده نکنید (مستقیماً از Bitmap تولیدی استفاده کنید) یا آنرا تغییر دهید. در کل متد ResizeImage هم بر اساس تولید تصویر با کیفیت بالا تنظیم شده.

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۲/۱۱/۱۱ ۱۶:۳۲

با سلام . روش دیگری که نیاز به نصب کتابخانه جانبی دیگر نداشته باشد و بتوان از آن تحت وب نیز استفاده کرد (برای هر دو ورژن x64 و x86) را چه توصیه میکنید؟ با تشکر.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۱۱ ۱۷:۰۶

[کتابخانه‌ی GhostScript هم هست](#) . یک سری ابزار دیگر هم [در اینجا](#) لیست شدند.