

How to extract images from PDF with PDF extractor SDK in ASP.NET C# with ByteScout Premium Suite

If you want to learn more then this tutorial will show how to extract images from PDF with PDF extractor SDK in ASP.NET C#.

The code displayed below will guide you to install an ASP.NET C# app to extract images from PDF with PDF extractor SDK. Want to extract images from PDF with PDF extractor SDK in your ASP.NET C# app? ByteScout Premium Suite is designed for it. ByteScout Premium Suite is the bundle that includes twelve SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording.

The SDK samples given below describe how to quickly make your application do extract images from PDF with PDF extractor SDK in ASP.NET C# with the help of ByteScout Premium Suite. Simply copy and paste in your ASP.NET C# project or application you and then run your app! Check ASP.NET C# sample code samples to see if they respond to your needs and requirements for the project.

Trial version of ByteScout Premium Suite is available for free. Source code samples are included to help you with your ASP.NET C# app.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout Premium Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Premium Suite](#)

[Get Free API key for Web API](#)

[visit \[www.ByteScout.com\]\(http://www.ByteScout.com\)](#)

Source Code Files:

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="ExtractImages.Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

        </div>
    </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Drawing.Imaging;
using System.IO;
using Bytescout.PDFExtractor;

namespace ExtractImages
{
    /*
     * IF YOU SEE TEMPORARY FOLDER ACCESS ERRORS:
     *
     * Temporary folder access is required for web application when you use ByteScout SDK.
     * If you are getting errors related to the access to temporary folder like "Access to the path ...
     *
     * SOLUTION:
     *
     * If your IIS Application Pool has "Load User Profile" option enabled the IIS provider ...
     *
     * If you are running Web Application under an impersonated account or IIS_IUSRS group ...
     *
     * In this case
     * - check the User or User Group your web application is running under
     * - then add permissions for this User or User Group to read and write into that temporary folder
     * - restart your web application and try again
     */
}
```

```

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        String inputFile = Server.MapPath(@".\bin\sample1.pdf");

        // Create Bytescout.PDFExtractor.ImageExtractor instance
        ImageExtractor extractor = new ImageExtractor();
        extractor.RegistrationName = "demo";
        extractor.RegistrationKey = "demo";

        // Load sample PDF document
        extractor.LoadDocumentFromFile(inputFile);

        // Array to keep extracted images
        List<byte[]> extractedImages = new List<byte[]>();

        // Initialize image enumeration
        if (extractor.GetFirstImage())
        {
            do
            {
                // Extract image to memory
                using (MemoryStream memoryStream = new MemoryStream())
                {
                    extractor.SaveCurrentImageToStream(memoryStream);
                    // Keep image as byte array
                    extractedImages.Add(memoryStream.ToArray());
                }
            }
            while (extractor.GetNextImage()); // Advance image enumeration
        }

        // Write first image to the output stream
        Response.Clear();

        Response.ContentType = "image/png";
        Response.AddHeader("Content-Disposition", "inline;filename=imag

        Response.BinaryWrite(extractedImages[0]);

        Response.End();
    }
}

```

Default.aspx.designer.cs

```
//-----
```

```

// <auto-generated>
//   This code was generated by a tool.
//   Runtime Version:2.0.50727.4952
//
//   Changes to this file may cause incorrect behavior and will be lost if
//   the code is regenerated.
// </auto-generated>
//-----
namespace ExtractAllImages {

    /// <summary>
    /// _Default class.
    /// </summary>
    /// <remarks>
    /// Auto-generated class.
    /// </remarks>
    public partial class _Default {

        /// <summary>
        /// form1 control.
        /// </summary>
        /// <remarks>
        /// Auto-generated field.
        /// To modify move field declaration from designer file to code-behind file.
        /// </remarks>
        protected global::System.Web.UI.HtmlControls.HtmlForm form1;
    }
}

```

ExtractImages.sln

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 2013
VisualStudioVersion = 12.0.40629.0
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ExtractImages", "ExtractImages.csproj"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Release|Any CPU.Build.0 = Release|Any CPU

```

```
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection
EndGlobal
```

Web.config

```
<?xml version="1.0"?>

<configuration>

    <appSettings/>
    <connectionStrings/>

    <system.web>
        <!--
            Set compilation debug="true" to insert debugging
            symbols into the compiled page. Because this
            affects performance, set this value to true only
            during development.
        -->
        <compilation debug="true" />
        <!--
            The <authentication> section enables configuration
            of the security authentication mode used by
            ASP.NET to identify an incoming user.
        -->
        <authentication mode="Windows" />
        <!--
            The <customErrors> section enables configuration
            of what to do if/when an unhandled error occurs
            during the execution of a request. Specifically,
            it enables developers to configure html error pages
            to be displayed in place of a error stack trace.
        -->
        <customErrors mode="RemoteOnly" defaultRedirect="GenericErrorPage.htm">
            <error statusCode="403" redirect="NoAccess.htm" />
            <error statusCode="404" redirect="NotFound.htm" />
        </customErrors>
        <!--
    </system.web>
</configuration>
```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Premium Suite Home Page](#)
[Explore ByteScout Premium Suite Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Premium Suite Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)
[Explore Web API Docs](#)
[Explore Web API Samples](#)

[visit www.ByteScout.com](#)

[visit www.PDF.co](#)

[www.bytescout.com](#)