

How to extract images from PDF with PDF extractor SDK in ASP.NET using ByteScout PDF Suite

Continuous learning is a crucial part of computer science and this tutorial shows how to extract images from PDF with PDF extractor SDK in ASP.NET

On this page you will learn from code samples for programming in ASP.NET. Writing of the code to extract images from PDF with PDF extractor SDK in ASP.NET can be executed by programmers of any level using ByteScout PDF Suite. Want to extract images from PDF with PDF extractor SDK in your ASP.NET app? ByteScout PDF Suite is designed for it. ByteScout PDF Suite is the set that includes 6 SDK products to work with PDF from generating rich PDF reports to extracting data from PDF documents and converting them to HTML. This bundle includes PDF (Generator) SDK, PDF Renderer SDK, PDF Extractor SDK, PDF to HTML SDK, PDF Viewer SDK and PDF Generator SDK for Javascript.

These ASP.NET code samples for ASP.NET guide developers to speed up coding of the application when using ByteScout PDF Suite. Simply copy and paste in your ASP.NET project or application you and then run your app! This basic programming language sample code for ASP.NET will do the whole work for you to extract images from PDF with PDF extractor SDK.

Our website gives trial version of ByteScout PDF Suite for free. It also includes documentation and source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Suite](#)

[Explore API Documentation](#)

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

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

[visit 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();
    }
}

```

```
//-----
// <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
EndGlobal
```

```
{0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Release|Any CPU.ActiveCfg = Rel
{0C256397-34FA-4067-98A7-01D3D2BE0F7E}.Release|Any CPU.Build.0 = Relas
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 PDF Suite Home Page](#)

[Explore ByteScout PDF Suite Documentation](#)

[Explore Samples](#)

[Sign Up for ByteScout PDF 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](#)