

How to extract attachments from PDF with PDF extractor SDK in ASP.NET C# using ByteScout Data Extraction Suite

Learn to code in ASP.NET C# to extract attachments from PDF with PDF extractor SDK with this step-by-step tutorial

The sample shows instructions and algorithm of how to extract attachments from PDF with PDF extractor SDK and how to make it run in your ASP.NET C# application. Want to extract attachments from PDF with PDF extractor SDK in your ASP.NET C# app? ByteScout Data Extraction Suite is designed for it. ByteScout Data Extraction Suite is the set that includes 3 SDK products for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK.

This prolific sample source code in ASP.NET C# for ByteScout Data Extraction Suite contains various functions and other necessary options you should do calling the API to extract attachments from PDF with PDF extractor SDK. Follow the instructions from scratch to work and copy the ASP.NET C# code. Enjoy writing a code with ready-to-use sample codes in ASP.NET C#.

If you want to try other source code samples then the free trial version of ByteScout Data Extraction Suite is available for download from our website. Just try other source code samples for ASP.NET C#.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Data Extraction Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Data Extraction 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="Ex
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xl

<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 Bytescout.PDFExtractor;

namespace ExtractAttachments
{
  public partial class _Default : System.Web.UI.Page
  {

    /*
    IF YOU SEE TEMPORARY FOLDER ACCESS ERRORS:

    Temporary folder access is required for web application when you use ByteScout
    If you are getting errors related to the access to temporary folder like "Access

    SOLUTION:

    If your IIS Application Pool has "Load User Profile" option enabled the IIS pro

    If you are running Web Application under an impersonated account or IIS_IUSRS g

    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
    - restart your web application and try again

    */
```

```

protected void Page_Load(object sender, EventArgs e)
{
    String inputFile = Server.MapPath(@"..\bin\attachments.pdf");

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

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

    Response.Clear();
    Response.ContentType = "text/html";

    for (int i = 0; i < extractor.Count; i++)
    {
        string attachmentFileName = extractor.GetFileName(i);
        int attachmentFileSize = extractor.GetSize(i);

        Response.Write("Found attachment: " + attachmentFileName + "<br/>");

        // You can save the attachment to a file
        //extractor.Save(i, attachmentFileName);

        // ... or write to output stream
        //extractor.Save(i, Response.OutputStream);

        Response.Write("- file size: " + attachmentFileSize + "<br/><br/>");
    }

    Response.End();

    extractor.Dispose();
}
}
}

```

Default.aspx.designer.cs

```

//-----
// <auto-generated>
// This code was generated by a tool.
// Runtime Version:2.0.50727.42
//
// Changes to this file may cause incorrect behavior and will be lost if
// the code is regenerated.
// </auto-generated>
//-----

```

```

namespace ExtractAttachments
{
    public partial class _Default
    {
        protected System.Web.UI.HtmlControls.HtmlForm form1;
    }
}

```

ExtractAttachments.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}") = "ExtractAttachments", "ExtractAttachments.csproj", "{...}"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {A8705DB2-82C7-46E7-8786-058EEE78DE9D}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {A8705DB2-82C7-46E7-8786-058EEE78DE9D}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {A8705DB2-82C7-46E7-8786-058EEE78DE9D}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {A8705DB2-82C7-46E7-8786-058EEE78DE9D}.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/>
</configuration>

```

```
<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" targetFramework="4.0"/>
  <!--
    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="FileNotFound.htm" />
    </customErrors>
  -->
  <pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID"/>
</system.web>
</configuration>
```

VIDEO

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

ON-PREMISE OFFLINE SDK

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