How to extract attachments from PDF in ASP.NET C# and ByteScout PDF Extractor SDK

The tutorial shows how to extract attachments from PDF in ASP.NET C#

These source code samples are listed and grouped by their programming language and functions they use. ByteScout PDF Extractor SDK is the SDK that helps developers to extract data from unstructured documents, pdf, images, scanned and electronic forms. Includes AI functions like automatic table detection, automatic table extraction and restructuring, text recognition and text restoration from pdf and scanned documents. Includes PDF to CSV, PDF to XML, PDF to JSON, PDF to searchable PDF functions as well as methods for low level data extraction. It can be used to extract attachments from PDF using ASP.NET C#.

Fast application programming interfaces of ByteScout PDF Extractor SDK for ASP.NET C# plus the instruction and the code below will help you quickly learn how to extract attachments from PDF. This ASP.NET C# sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Test ASP.NET C# sample code examples whether they respond your needs and requirements for the project.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout PDF Extractor SDK

Explore API Documentation

Get Free Training for ByteScout PDF Extractor SDK

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="E

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

<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 "Acce

        SOLUTION:

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

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

        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", "ExtractAtta
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
		{A8705DB2-82C7-46E7-8786-058EEE78DE9D}.Debug|Any CPU.Build.0 = Debug|A
		{A8705DB2-82C7-46E7-8786-058EEE78DE9D}.Release|Any CPU.ActiveCfg = Rele
		{A8705DB2-82C7-46E7-8786-058EEE78DE9D}.Release|Any CPU.Build.0 = Releas
	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" 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=s28W3_KMraU

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout PDF Extractor SDK Home Page
Explore ByteScout PDF Extractor SDK Documentation
Explore Samples
Sign Up for ByteScout PDF Extractor SDK Online Training

ON-DEMAND REST WEB API