

www.bytescout.com

batch process PDF to text in ASP.NET C# with ByteScout PDF Extractor SDK

Tutorial: how to do batch process PDF to text in ASP.NET C#

Today you are going to learn how to batch process PDF to text in ASP.NET C#. Batch process PDF to text in ASP.NET C# can be implemented with ByteScout PDF Extractor SDK. ByteScout PDF Extractor SDK is the SDK designed to help developers with pdf tables and pdf data extraction from unstructured documents like pdf, tiff, scans, images, scanned and electronic forms. The library is powered by OCR, computer vision and AI to provide unique functionality like table detection, automatic table structure extraction, data restoration, data restructuring and reconstruction. Supports PDF, TIFF, PNG, JPG images as input and can output CSV, XML, JSON formatted data. Includes full set of utilities like pdf splitter, pdf merger, searchable pdf maker and other utilities.

The SDK samples like this one below explain how to quickly make your application do batch process PDF to text in ASP.NET C# with the help of ByteScout PDF Extractor SDK. This ASP.NET C# sample code should be copied and pasted into your application's code editor. Then just compile and run it to see how it works. Enjoy writing a code with ready-to-use sample ASP.NET C# codes to implement batch process PDF to text using ByteScout PDF Extractor SDK.

Our website provides free trial version of ByteScout PDF Extractor SDK. It comes along with all these source code samples with the goal to help you with your ASP.NET C# application implementation.

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

Source Code Files:

BatchProcessing.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}") = "BatchProcessing", "BatchProcessing"
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
```

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="BatchProcessing.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.IO;
using Bytescout.PDFExtractor;

namespace BatchProcessing
{
    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 PDF Extractor component. If you are getting errors related to the access to temporary folder like "Access to the path 'C:\Windows\Temp\Temporary Internet Files\Content.IE5\...'" then you need to grant temporary folder access to your application. To do this, go to IIS Manager, select your application, click "Edit Application" button, then "Advanced Settings" link under "Application Settings" section. In "Application Pool" section, click "Select Application Pool..." button and choose your application pool. Then click "Edit..." button next to "Temporary ASP.NET files" setting. In "Temporary ASP.NET files" dialog, set "Physical Path" to a folder where your application can write (e.g. "C:\inetpub\temp\Temporary ASP.NET Files\BatchProcessing\"). Click "OK" button to close all dialogs and save changes.

        SOLUTION:

        If your IIS Application Pool has "Load User Profile" option enabled the IIS process will run under your application pool identity. If you are running Web Application under an impersonated account or IIS_IUSRS group, then you need to grant temporary folder access to this account/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 folder
        - restart your web application and try again

        */

        protected void Page_Load(object sender, EventArgs e)
        {
            // Directory containing test files
            String inputFolder = Server.MapPath(@".\bin");

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

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

            // Get PDF files
            string[] pdfFiles = Directory.GetFiles(inputFolder, "*.pdf");

            foreach (string file in pdfFiles)
            {
                // Load document
                extractor.LoadDocumentFromFile(file);

                Response.Output.Write("<pre>");
                Response.Output.Write(extractor.GetText());
                Response.Output.Write("</pre>");
            }
        }
    }
}
```

```
// Extract document text and write to the output stream
extractor.SaveTextToStream(Response.OutputStream);

Response.Output.Write("</pre>");
Response.Output.Write("<br/>");

// Reset the extractor before loading another file
extractor.Reset();
}

Response.End();
}

}

}
```

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 BatchProcessing
{

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

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="NotFound.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

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

[visit www.ByteScout.com](#)

[visit www.PDF.co](#)

[www.bytescout.com](#)