

www.bytescout.com

OCR (optical character recognition) in PDF with PDF extractor SDK in ASP.NET and ByteScout PDF Suite

Learn to code OCR (optical character recognition) in PDF with PDF extractor SDK in ASP.NET: How-To tutorial

The samples of source code documentation give a quick and simple method to apply a required functionality into your application. ByteScout PDF Suite helps with OCR (optical character recognition) in PDF with PDF extractor SDK in ASP.NET. ByteScout PDF Suite is the bundle that provides six different SDK libraries 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.

If you want to quickly learn then these fast application programming interfaces of ByteScout PDF Suite for ASP.NET plus the guideline and the ASP.NET code below will help you quickly learn OCR (optical character recognition) in PDF with PDF extractor SDK. Follow the steps-by-step instructions from the scratch to work and copy and paste code for ASP.NET into your editor. ASP.NET application implementation mostly involves various stages of the software development so even if the functionality works please check it with your data and the production environment.

Our website gives free trial version of ByteScout PDF Suite. It includes all these source code samples with the purpose to assist you with your ASP.NET application implementation.

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="Opc<br/><!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 Bytescout.PDFExtractor;  
  
// Before running the example, copy missing *.traineddata files from "Redistributable"  
// or download from https://github.com/bytescout/ocrdata  
// Make sure "Copy to Output Directory" property of each added language file is set to  
namespace OpticalCharacterRecognition  
{  
  
    /*  
     * 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  
     *  
     * 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 temp
- 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\sample_ocr.pdf");

        // Location of language files
        String ocrLanguageDataFolder = Server.MapPath(@".\bin\ocrdata");

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

            // Enable Optical Character Recognition (OCR)
            // in .Auto mode (SDK automatically checks if needs to use OCR
            extractor.OCRMode = OCRMode.Auto;
            // Set the location of OCR language data files
            extractor.OCRLanguageDataFolder = ocrLanguageDataFolder;
            // Set OCR language
            extractor.OCRLanguage = "eng"; // "eng" for english, "deu" for
            // Set PDF document rendering resolution
            extractor.OCRResolution = 300;

            // You can also apply various preprocessing filters
            // to improve the recognition on low-quality scans.

            // Automatically deskew skewed scans
            //extractor.OCRImagePreprocessingFilters.AddDeskew();

            // Remove vertical or horizontal lines (sometimes helps to avoid
            //extractor.OCRImagePreprocessingFilters.AddVerticalLinesRemove();
            //extractor.OCRImagePreprocessingFilters.AddHorizontalLinesRemove();

            // Repair broken letters
            //extractor.OCRImagePreprocessingFilters.AddDilate();

            // Remove noise
            //extractor.OCRImagePreprocessingFilters.AddMedian();

            // Apply Gamma Correction
            //extractor.OCRImagePreprocessingFilters.AddGammaCorrection();

            // Add Contrast
            //extractor.OCRImagePreprocessingFilters.AddContrast(20);

            // (!) You can use new OCRAnalyser class to find an optimal
            // filters for your specific document.
            // See "OCR Analyser" example.

            // Load PDF document
        }
    }
}
```

```
    extractor.LoadDocumentFromFile(inputFile);

    // Write extracted text to output stream
    Response.Clear();
    Response.ContentType = "text/html";

    Response.Write("<pre>");
    // Write extracted text to output stream
    Response.Write(extractor.GetText());
    Response.Write("</pre>");

    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 ExtractAllText {

    /// <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;
    }
}
```

OpticalCharacterRecognition.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}") = "OpticalCharacterRecognition", "OpticalCharacterRecognition"
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
    GlobalSection(ExtensibilityGlobals) = postSolution
        SolutionGuid = {0C72D936-8833-4995-B5F5-FFCC9B15733B}
    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](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com