

www.bytescout.com

How to extract text from page area in PDF with PDF extractor SDK in ASP.NET C# using ByteScout Premium Suite

Step-by-step tutorial on how to extract text from page area in PDF with PDF extractor SDK in ASP.NET C#

The sample shows instructions and algorithm of how to extract text from page area in PDF with PDF extractor SDK and how to make it run in your ASP.NET C# application. ByteScout Premium Suite can extract text from page area in PDF with PDF extractor SDK. It can be applied from ASP.NET C#. ByteScout Premium Suite is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording.

These ASP.NET C# code samples for ASP.NET C# guide developers to speed up coding of the application when using ByteScout Premium Suite. Follow the instructions from scratch to work and copy the ASP.NET C# code. Applying ASP.NET C# application mostly includes various stages of the software development so even if the functionality works please test it with your data and the production environment.

Trial version of ByteScout Premium Suite is available for free. Source code samples are included to help you with your ASP.NET C# app.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Premium Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Premium 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="ExtractTextFromPageArea._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.Drawing;
using Bytescout.PDFExtractor;

namespace ExtractTextFromPageArea
{
    /*
    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 temporary folder is denied" or "The process cannot access the file because it is being used by another process", please make sure that your application has sufficient permissions to write to the temporary folder.

    SOLUTION:

    If your IIS Application Pool has "Load User Profile" option enabled the IIS provider will run under the context of the application pool user account. In this case you need to grant write permission to the temporary folder for the application pool user account.

    If you are running Web Application under an impersonated account or IIS_IUSRS group, you need to grant write permission to the temporary folder for the impersonated user 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\columns.pdf");

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

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

    // Get dimensions of the first document page
    RectangleF rectangle = extractor.GetPageRectangle(0);

    // Get text from the 1/3 of the page
    rectangle.Width = rectangle.Width / 3f;

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

    extractor.SetExtractionArea(rectangle);

    Response.Write("<pre>");

    // Save extracted text to output stream
    extractor.SavePageTextToStream(0, Response.OutputStream);

    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;
}

}
```

ExtractTextFromPageArea.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}") = "ExtractTextFromPageArea", "ExtractTextFromPageArea"
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
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 Premium Suite Home Page](#)

[Explore ByteScout Premium Suite Documentation](#)

[Explore Samples](#)

[Sign Up for ByteScout Premium 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](#)