www.bytescout.com

How to decode page by page from PDF OR TIFF in ASP.NET C# and ByteScout BarCode Reader SDK

This code in ASP.NET C# shows how to decode page by page from PDF OR TIFF with this how to tutorial

Source code documentation samples provide quick and easy way to add a required functionality into your application. ByteScout BarCode Reader SDK is the barcode decoder with support for code 39, code 128, QR Code, Datamatrix, GS1, PDF417 and all other popular barcodes. Can read barcodes from images, pdf, tiff documents and live web camera. Supports noisy and damaged documents, can split and merge pdf and tiff documents based on barcodes. Can export barcode decoder results to XML, JSON, CSV and into custom data structures and you can use it to decode page by page from PDF OR TIFF with ASP.NET C#.

This code snippet below for ByteScout BarCode Reader SDK works best when you need to quickly decode page by page from PDF OR TIFF in your ASP.NET C# application. In order to implement the functionality, you should copy and paste this code for ASP.NET C# below into your code editor with your app, compile and run your application. Further enhancement of the code will make it more vigorous.

Trial version of ByteScout BarCode Reader SDK can be downloaded for free from our website. It also includes source code samples for ASP.NET C# and other programming languages.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout BarCode Reader SDK

Explore API Documentation

Get Free Training for ByteScout BarCode Reader 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="We

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

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Web Barcode Reader Tester (C#)</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Click browse button to upload a PDF/TIFF document<br />
            <asp:FileUpload ID="FileUpload1" runat="server" /><br />
            <br />
            Page No. to read barcode from:
        <asp:TextBox ID="txtPageNo" Text="1" runat="server"></asp:TextBox>
            <br />
            <br />
            <asp:Button ID="UploadButton"
                Text="Upload file and read barcodes"
                OnClick="UploadButton_Click"
                runat="server"></asp:Button>
            <br />
            <asp:Label ID="UploadStatusLabel" Text="" runat="server"></asp:Label>
            <br />
            <asp:ListBox ID="ListBox1" runat="server" Visible="False"></asp:ListBox><br
            <br />
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using Bytescout.BarCodeReader;

namespace WebTestSharp
{
        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 "Acces

        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 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 UploadButton_Click(object sender, EventArgs e)
                {
                        String savePath = @"\uploads\";

                        if (FileUpload1.HasFile)
                        {
                                String fileName = FileUpload1.FileName;
                                savePath += fileName;
                                FileUpload1.SaveAs(Server.MapPath(savePath));

                                Reader barcodeReader = new Reader();

                                // Limit search to 1D barcodes only (exclude 2D barcode
                                // Change to barcodeReader.BarcodeTypesToFind.SetAll()
                                // or select specific type, e.g. barcodeReader.Barcode
                                barcodeReader.BarcodeTypesToFind.SetAll1D();

                                // reader.MediumTrustLevelCompatible = true; // uncomm
                                UploadStatusLabel.Visible = false;

                // Page No to read
                int PageNoToRead = Convert.ToInt32(txtPageNo.Text);

                ListBox1.Items.Clear();
                                ListBox1.Visible = true;
                                ListBox1.Items.Add("Reading barcode(s) from file \"" +

                // Reading barcode from document on page-by-page basic
                                FoundBarcode[] barcodes = barcodeReader.ReadFrom(Serve

                                if (barcodes.Length == 0)
                                {
                                        ListBox1.Items.Add("No barcodes found");
                                }
                                else
                                {
                                        foreach (FoundBarcode barcode in barcodes)
                                        {
                        ListBox1.Items.Add(String.Format("Found barcode - Type: '{0}',
                                        }
                                }
                        }
                        else
                        {
```

```
                                // Notify the user that a file was not uploaded.
                                UploadStatusLabel.Text = "You did not specify a file to
                            }
                        }
                    }
}
```

Default.aspx.designer.cs

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

namespace WebTestSharp {


    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;

        /// <summary>
        /// FileUpload1 control.
        /// </summary>
        /// <remarks>
        /// Auto-generated field.
        /// To modify move field declaration from designer file to code-behind file.
        /// </remarks>
        protected global::System.Web.UI.WebControls.FileUpload FileUpload1;

        /// <summary>
        /// txtPageNo control.
        /// </summary>
        /// <remarks>
        /// Auto-generated field.
        /// To modify move field declaration from designer file to code-behind file.
        /// </remarks>
        protected global::System.Web.UI.WebControls.TextBox txtPageNo;
```

```
        /// <summary>
        /// UploadButton control.
        /// </summary>
        /// <remarks>
        /// Auto-generated field.
        /// To modify move field declaration from designer file to code-behind file.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Button UploadButton;

        /// <summary>
        /// UploadStatusLabel control.
        /// </summary>
        /// <remarks>
        /// Auto-generated field.
        /// To modify move field declaration from designer file to code-behind file.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Label UploadStatusLabel;

        /// <summary>
        /// ListBox1 control.
        /// </summary>
        /// <remarks>
        /// Auto-generated field.
        /// To modify move field declaration from designer file to code-behind file.
        /// </remarks>
        protected global::System.Web.UI.WebControls.ListBox ListBox1;
    }
}
```

Web.config

```
<?xml version="1.0"?>

<configuration>

    <appSettings/>
    <connectionStrings/>

    <system.web>
      <httpRuntime maxRequestLength="1048576" />

      <!--
            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="FileNotFound.htm" />
        </customErrors>
        -->
    </system.web>
</configuration>
```

VIDEO

https://www.youtube.com/watch?v=EARSPJFIJMU

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout BarCode Reader SDK Home Page
Explore ByteScout BarCode Reader SDK Documentation
Explore Samples
Sign Up for ByteScout BarCode Reader 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