

[www.bytescout.com](http://www.bytescout.com)

## How to read barcodes from pdf with barcode reader sdk in ASP.NET MVC C# and ByteScout Data Extraction Suite

How to write a robust code in ASP.NET MVC C# to read barcodes from pdf with barcode reader sdk with this step-by-step tutorial

We made thousands of pre-made source code pieces for easy implementation in your own programming projects. ByteScout Data Extraction Suite is the bundle that includes three SDK tools for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK. It can be applied to read barcodes from pdf with barcode reader sdk using ASP.NET MVC C#.

These ASP.NET MVC C# code samples for ASP.NET MVC C# guide developers to speed up coding of the application when using ByteScout Data Extraction Suite. Simply copy and paste in your ASP.NET MVC C# project or application you and then run your app! Enjoy writing a code with ready-to-use sample ASP.NET MVC C# codes.

If you want to try other source code samples then the free trial version of ByteScout Data Extraction Suite is available for download from our website. Just try other source code samples for ASP.NET MVC C#.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Data Extraction Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Data Extraction 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="WebBarcodeReaderTester.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>Web Barcode Reader Tester (C#)</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Click browse button to upload a PDF document<br />
            <asp:FileUpload ID="FileUpload1" runat="server" /><br />
            <br />
            <asp:Button id="UploadButton"
                Text="Upload file"
                OnClick="UploadButton_Click"
                runat="server">
            </asp:Button>
            <br />
            <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
{
    /*
     * 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"

     SOLUTION:
     If your IIS Application Pool has "Load User Profile" option enabled the IIS provider
     will have access to temporary folder
    */
}
```

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 temporary folder
- restart your web application and try again

\*/

```
public partial class _Default : System.Web.UI.Page
{
    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.BarcodeTypesToFind.SetAll1D();

            // reader.MediumTrustLevelCompatible = true; // uncomment if needed
            UploadStatusLabel.Visible = false;

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

            FoundBarcode[] barcodes = barcodeReader.ReadFrom(Server.MapPath(savePath));

            if (barcodes.Length == 0)
            {
                ListBox1.Items.Add("No barcodes found");
            }
            else
            {
                foreach (FoundBarcode barcode in barcodes)
                {
                    ListBox1.Items.Add(String.Format("Found barcode with type '{0}'\n{1}", barcode.Type, barcode.Data));
                }
            }
        }
        else
        {
            // Notify the user that a file was not uploaded.
            UploadStatusLabel.Text = "You did not specify a file to upload";
        }
    }
}
```

## Default.aspx.designer.cs

```
//----------------------------------------------------------------------------  
// <auto-generated>  
//   This code was generated by a tool.  
//   Runtime Version:2.0.50727.4927  
//  
//   Changes to this file may cause incorrect behavior and will be lost if  
//   the code is regenerated.  
// </auto-generated>  
//----------------------------------------------------------------------------  
  
namespace WebTestSharp {  
  
    /// <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;  
  
        /// <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>  
        /// 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>
```

```

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

---

## VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

## ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Data Extraction Suite Home Page](#)  
[Explore ByteScout Data Extraction Suite Documentation](#)  
[Explore Samples](#)  
[Sign Up for ByteScout Data Extraction 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](#)