

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

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

Read barcodes from image with barcode reader sdk is simple to apply in ASP.NET MVC C# if you use these source codes below. ByteScout Data Extraction Suite can read barcodes from image with barcode reader sdk. It can be applied from ASP.NET MVC C#. ByteScout Data Extraction Suite is the set that includes 3 SDK products for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK.

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. Just copy and paste the code into your ASP.NET MVC C# application's code and follow the instructions. Applying ASP.NET MVC 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.

You can download free trial version of ByteScout Data Extraction Suite from our website to see and try many others 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="WebTestSharp.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 an image<br />
            <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 />
        <asp:Image ID="Image1" runat="server" Height="74px" Width="71px" /></div>
    </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Drawing;
using System.IO;
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 path is denied"
SOLUTION:

If your IIS Application Pool has "Load User Profile" option enabled the IIS provide

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

        if (FileUpload1.HasFile)
        {
            string virtualFilePath = Path.Combine(savePath, FileUpload1.FileName);

            string serverFilePath = Server.MapPath(virtualFilePath);
            if (!Directory.Exists(Path.GetDirectoryName(serverFilePath)))
                Directory.CreateDirectory(Path.GetDirectoryName(serverFilePath));

            FileUpload1.SaveAs(serverFilePath);

            Image image = null;

            try
            {
                using (Stream fileStream = File.OpenRead(serverFilePath))
                {
                    image = Image.FromStream(fileStream);
                }
            }
            catch (Exception)
            {
            }

            if (image == null)
            {
                UploadStatusLabel.Visible = true;
                UploadStatusLabel.Text = "Your file is not an image";
                Image1.Visible = false;
                ListBox1.Visible = false;
            }
            else
            {
                UploadStatusLabel.Visible = false;
                Image1.ImageUrl = virtualFilePath;
                Image1.Visible = true;
                Image1.Width = image.Width;
                Image1.Height = image.Height;
                ListBox1.Items.Clear();
                ListBox1.Visible = true;
            }
        }
    }
}
```

```

        FindBarcodes(serverFilePath);

        if (ListBox1.Items.Count == 0)
            ListBox1.Items.Add("No barcodes found");
    }
}
else
{
    // Notify the user that a file was not uploaded.
    UploadStatusLabel.Text = "You did not specify a file to upload";
}
}

private void FindBarcodes(string fileName)
{
    Reader reader = new Reader();

    // Limit search to 1D barcodes only (exclude 2D barcodes to speed up search)
    // Change to reader.BarcodeTypesToFind.SetAll() to scan for all types
    // or select specific type, e.g. reader.BarcodeTypesToFind.PDF417
    reader.BarcodeTypesToFind.SetAll1D();

    // reader.MediumTrustLevelCompatible = true; // uncomment this if you are using .NET Framework 4.0
    reader.ReadFromFile(fileName);

    foreach (FoundBarcode barcode in reader.FoundBarcodes)
    {
        ListBox1.Items.Add(String.Format("{0} : {1}", barcode.Type, barcode.Value));
    }
}
}
}

```

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>
    /// 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;

    /// <summary>
    /// Image1 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.Image Image1;

```

```

    }

```

```

}

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;

namespace Barcodes_From_Image
{
    public class Global : System.Web.HttpApplication
    {
        void Application_Start(object sender, EventArgs e)
        {
            // Code that runs on application startup
        }

        void Application_End(object sender, EventArgs e)
        {
            // Code that runs on application shutdown
        }

        void Application_Error(object sender, EventArgs e)
        {
            // Code that runs when an unhandled error occurs
        }

        void Session_Start(object sender, EventArgs e)
        {
            // Code that runs when a new session is started
        }

        void Session_End(object sender, EventArgs e)
        {
            // Code that runs when a session ends.
            // Note: The Session_End event is raised only when the sessionstate mode
            // is set to InProc in the Web.config file. If session mode is set to State
            // or SQLServer, the event is not raised.
        }
    }
}
```

Site.Master.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Barcodes_From_Image
{
    public partial class SiteMaster : System.Web.UI.MasterPage
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Site.Master.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 Barcodes_From_Image
{

    public partial class SiteMaster
    {

        /// <summary>
        /// HeadContent 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.ContentPlaceHolder HeadContent;

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

/// <summary>
/// HeadLoginView 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.LoginView HeadLoginView;

/// <summary>
/// HeadHomeLink 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.HtmlAnchor HeadHomeLink;

/// <summary>
/// HeadAboutLink 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.HtmlAnchor HeadAboutLink;

/// <summary>
/// MainContent 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.ContentPlaceHolder MainContent;
}
}

```



```

<?xml version="1.0"?>

<!-- For more information on using web.config transformation visit http://go.microsoft

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">
  <!--
    In the example below, the "SetAttributes" transform will change the value of
    "connectionString" to use "ReleaseSQLServer" only when the "Match" locator
    finds an attribute "name" that has a value of "MyDB".

    <connectionStrings>
      <add name="MyDB"
        connectionString="Data Source=ReleaseSQLServer;Initial Catalog=MyReleaseDB;Int
        xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>
    </connectionStrings>
  -->
  <system.web>
    <!--
      In the example below, the "Replace" transform will replace the entire
      <customErrors> section of your web.config file.
      Note that because there is only one customErrors section under the
      <system.web> node, there is no need to use the "xdt:Locator" attribute.

      <customErrors defaultRedirect="GenericError.htm"
        mode="RemoteOnly" xdt:Transform="Replace">
        <error statusCode="500" redirect="InternalError.htm"/>
      </customErrors>
    -->
  </system.web>
</configuration>

```

Web.Release.config

```

<?xml version="1.0"?>

<!-- For more information on using web.config transformation visit http://go.microsoft

<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">
  <!--
    In the example below, the "SetAttributes" transform will change the value of
    "connectionString" to use "ReleaseSQLServer" only when the "Match" locator
    finds an attribute "name" that has a value of "MyDB".

    <connectionStrings>
      <add name="MyDB"
        connectionString="Data Source=ReleaseSQLServer;Initial Catalog=MyReleaseDB;Int
        xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>
    </connectionStrings>
  -->
  <system.web>

```

```

<compilation xdt:Transform="RemoveAttributes(debug)" />
<!--
  In the example below, the "Replace" transform will replace the entire
  <customErrors> section of your web.config file.
  Note that because there is only one customErrors section under the
  <system.web> node, there is no need to use the "xdt:Locator" attribute.

  <customErrors defaultRedirect="GenericError.htm"
    mode="RemoteOnly" xdt:Transform="Replace">
    <error statusCode="500" redirect="InternalError.htm"/>
  </customErrors>
-->
</system.web>
</configuration>

```

Web.config

```

<?xml version="1.0"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->

<configuration>
  <connectionStrings>
    <add name="ApplicationServices"
      connectionString="data source=.\SQLEXPRESS;Integrated Security=SSPI;AttachDBF=
      providerName="System.Data.SqlClient" />
    </connectionStrings>

    <system.web>
      <compilation debug="true" targetFramework="4.0" />

      <authentication mode="Forms">
        <forms loginUrl="~/Account/Login.aspx" timeout="2880" />
      </authentication>

      <membership>
        <providers>
          <clear/>
          <add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembership
            enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestion
            maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6" minRequiredNo
            applicationName="/" />
          </providers>
        </membership>

        <profile>
          <providers>
            <clear/>

```

```
<add name="AspNetSqlProfileProvider" type="System.Web.Profile.SqlProfileProvider" />
</providers>
</profile>

<roleManager enabled="false">
  <providers>
    <clear/>
    <add name="AspNetSqlRoleProvider" type="System.Web.Security.SqlRoleProvider" connectionStringName="LocalSqlServer" />
    <add name="AspNetWindowsTokenRoleProvider" type="System.Web.Security.WindowsTokenRoleProvider" />
  </providers>
</roleManager>

</system.web>

<system.webServer>
  <modules runAllManagedModulesForAllRequests="true"/>
</system.webServer>
</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](http://www.ByteScout.com)

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

