

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

## How to read barcode with barcode reader sdk in ASP.NET MVC C# and ByteScout Premium Suite

This code in ASP.NET MVC C# shows how to read barcode with barcode reader sdk with this how to tutorial

An easy to understand guide on how to read barcode with barcode reader sdk in ASP.NET MVC C# with this source code sample. Want to read barcode with barcode reader sdk in your ASP.NET MVC C# app? ByteScout Premium Suite is designed for it. ByteScout Premium Suite is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording.

The SDK samples given below describe how to quickly make your application do read barcode with barcode reader sdk in ASP.NET MVC C# with the help of ByteScout Premium Suite. Follow the instructions from scratch to work and copy the ASP.NET MVC C# code. Further improvement of the code will make it more robust.

Trial version of ByteScout Premium Suite is available for free. Source code samples are included to help you with your ASP.NET MVC 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="SimpleWebTestSharp._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>Simple Web Barcode Reader Tester (C#)</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <div>
                <br />
                <asp:ListBox ID="ListBox1" runat="server"></asp:ListBox><br />
                <br />
                <asp:Image ID="Image1" runat="server" />&nbsp;</div>
            </div>
        </form>
    </body>
</html>
```

## Default.aspx.cs

```
using System;
using Bytescout.BarCodeReader;

namespace SimpleWebTestSharp
{
    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 BarCode Reader component. If you are getting errors related to the access to temporary folder like "Access is denied" or "The process cannot access the file because it is being used by another process", please make sure that your IIS Application Pool has "Load User Profile" option enabled. If you are running Web Application under an impersonated account or IIS_IUSRS group, then add permissions for this User or User Group to read and write into that folder.

        SOLUTION:

        If your IIS Application Pool has "Load User Profile" option enabled the IIS process will run under the context of the user who is currently logged in. If you are running Web Application under an impersonated account or IIS_IUSRS group, then add permissions for this User or User Group to read and write into that folder.

        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 folder.
    }
}
```

```

- restart your web application and try again

*/
protected void Page_Load(object sender, EventArgs e)
{
    Image1.ImageUrl = "BarcodePhoto.jpg";

    Reader reader = new Reader();

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

    // reader.MediumTrustLevelCompatible = true; // uncomment this if you are running in medium trust

    reader.ReadFromFile(Server.MapPath("BarcodePhoto.jpg"));

    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.
//   Runtime Version:2.0.50727.3053
//
//   Changes to this file may cause incorrect behavior and will be lost if
//   the code is regenerated.
// </auto-generated>
//-----

namespace SimpleWebTestSharp {

    /// <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>
/// 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;
}

}

```

## 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](#)

