

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

Learn to code in ASP.NET MVC C# to batch read barcodes to csv with barcode reader sdk with this step-by-step tutorial

On this page you will learn from code samples for programming in ASP.NET MVC C#. Writing of the code to batch read barcodes to csv with barcode reader sdk in ASP.NET MVC C# can be executed by programmers of any level using ByteScout Data Extraction Suite. 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. It can be applied to batch read barcodes to csv 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. Follow the instructions from scratch to work and copy the ASP.NET MVC C# code. 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](#)

Source Code Files:

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs" Inherits="ImagePDFBarcodeToCSV.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Scan Image/PDF files for barcode and generate CSV</title>
</head>
<body>
    <form id="form1" runat="server">

        <label>
            This project demonstrates scanning of folder, and processing each image/PDF file.
            It'll then generates CSV file containing barcodes and downloads it.
        </label>
        <br />
        <br />
        <div>
            <asp:Button ID="btnImagePDFBarcodeToCSV" Text="Scan for Barcode and Generate CSV" runat="server" OnClick="btnImagePDFBarcodeToCSV_Click" />
            <br />
            <br />
            <asp:Label ID="lblInfo" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs

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

namespace ImagePDFBarcodeToCSV
{
    /*
     * 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
     * 
```

SOLUTION:

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

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 Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnImagePDFBarcodeToCSV_Click(object sender, EventArgs e)
    {
        //Read Barcode Process
        Reader reader = new Reader();
        reader.RegistrationKey = "demo";
        reader.RegistrationName = "demo";

        // Set Barcode type to find
        reader.BarcodeTypesToFind.All = true;

        // Output list
        List<CSVOutputFormat> lstCSVOutput = new List<CSVOutputFormat>();

        // Get all files in folder, and iterate through each file
        var files = System.IO.Directory.GetFiles(Request.MapPath("~/BarcodeFiles"));
        foreach (var fileName in files)
        {
            // Read barcodes
            FoundBarcode[] barcodes = reader.ReadFrom(fileName);

            foreach (FoundBarcode code in barcodes)
            {
                lstCSVOutput.Add(new CSVOutputFormat { barcodeValue = code.Value, });
            }
        }

        // Set info text
        lblInfo.Text = string.Format("Total {0} barcode found in {1} file.", lstCSVOutput.Count);

        // Export to CSV
        ExportToCsv(lstCSVOutput, Response);
    }

    /// <summary>
    /// Exports to CSV
    /// </summary>
    /// <param name="lstCSVOutput"></param>
    private static void ExportToCsv(List<CSVOutputFormat> lstCSVOutput, HttpResponse response)
    {
        response.ContentType = "text/csv";
        response.ContentEncoding = Encoding.UTF8;
        response.CacheControl = "no-cache";
        response.AddHeader("Content-Disposition", "attachment; filename=BarcodeReport.csv");
        response.BinaryWrite(Encoding.UTF8.GetBytes("Barcode Value" + "," + "Barcode Type" + "\r\n"));

        foreach (var item in lstCSVOutput)
        {
            response.BinaryWrite(Encoding.UTF8.GetBytes(item.barcodeValue + "," + item.type + "\r\n"));
        }
    }
}
```

```

    }

    System.Text.StringBuilder csvOutputContent = new System.Text.StringBuilder();

    csvOutputContent.Append("Barcode Value,Barcode Type,Scan DateTime,File Name");

    foreach (var item in lstCSVOutput)
    {
        csvOutputContent.AppendFormat("\r\n{0},{1},{2},{3}", item.barcodeValue,
        item.barcodeType, item.scanDateTime, item.fileName);
    }

    // Perform download of file
    response.Clear();
    response.ClearHeaders();

    response.AppendHeader("Content-Length", csvOutputContent.Length.ToString());
    response.ContentType = "text/csv";
    response.AppendHeader("Content-Disposition", "attachment;filename=\"output.csv\"");

    response.Write(csvOutputContent.ToString());
    response.End();
}

}

class CSVOutputFormat
{
    public string barcodeValue { get; set; }
    public string barcodeType { get; set; }
    public string scanDateTime { get; set; }
    public string fileName { get; set; }
}

}

```

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 ImagePDFBarcodeToCSV {

    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>
    /// btnImagePDFBarcodeToCSV 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 btnImagePDFBarcodeToCSV;

    ///<summary>
    /// lblInfo 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 lblInfo;
}

}

```

ImagePDFBarcodeToCSV.sln

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.27703.2026
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "ImagePDFBarcodeToCSV", "ImagePDFBarcodeToCSV.csproj", {2284DD7C-8475-49DF-B6C9-57F726573AA8}
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {2284DD7C-8475-49DF-B6C9-57F726573AA8}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
        {2284DD7C-8475-49DF-B6C9-57F726573AA8}.Debug|Any CPU.Build.0 = Debug|Any CPU
        {2284DD7C-8475-49DF-B6C9-57F726573AA8}.Release|Any CPU.ActiveCfg = Release|Any CPU
        {2284DD7C-8475-49DF-B6C9-57F726573AA8}.Release|Any CPU.Build.0 = Release|Any CPU
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE
    EndGlobalSection

```

```
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {27B3CAC5-CEAC-4F05-85BE-10B502086D8C}
EndGlobalSection
EndGlobal
```

Web.config

```
<?xml version="1.0" encoding="utf-8"?>

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

<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0" />
  </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](#)