

How to make slideshow from uploaded files (vb) in ASP.NET and ByteScout Image To Video SDK

How to make slideshow from uploaded files (vb) in ASP.NET

With this source code sample you may quickly learn how to make slideshow from uploaded files (vb) in ASP.NET. What is ByteScout Image To Video SDK? It is the library for conversion of images into video slideshow. Provides built-in support for more than hundred of 2D and 3D transition effects. Output formats supported are AVI, WMV and WEBM video files. You may adjust output video size, quality, framerate and add audio. It can help you to make slideshow from uploaded files (vb) in your ASP.NET application.

You will save a lot of time on writing and testing code as you may just take the ASP.NET code from ByteScout Image To Video SDK for make slideshow from uploaded files (vb) below and use it in your application. In order to implement the functionality, you should copy and paste this code for ASP.NET below into your code editor with your app, compile and run your application. Implementing ASP.NET application typically includes multiple stages of the software development so even if the functionality works please test it with your data and the production environment.

Free trial version of ByteScout Image To Video SDK is available on our website. Documentation and source code samples are included.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Image To Video SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Image To Video SDK](#)

[Get Free API key for Web API](#)

[visit www.Bytescout.com](http://www.Bytescout.com)

Source Code Files:

Default.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="Default.aspx.vb" Inherits="..." %>

<!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>Slideshow from uploaded images</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            Click browse button to upload an image:<br />
            <br />
            <asp:FileUpload ID="FileUpload1" runat="server" Width="255px" />
            <br />
            <br />
            <asp:Button ID="ButtonUpload" runat="server" Text="Upload" Width="90px" />
            <br />
            <br />
            Uploaded images:<br />
            <asp:ListBox ID="ListBoxUploadedImages" runat="server" Height="137px"
                Width="258px"></asp:ListBox>
            <br />
            <br />
            <asp:Button ID="ButtonMakeVideo" runat="server" Text="MakeVideo"
                Width="257px" />

        </div>
    </form>
</body>
</html>
```

Default.aspx.designer.vb

```
'-----
' <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>
'-----
```

Option Strict On
Option Explicit On

Partial Public Class _Default

```
'''<summary>  
'''form1 control.  
'''</summary>  
'''<remarks>  
'''Auto-generated field.  
'''To modify move field declaration from designer file to code-behind file.  
'''</remarks>
```

```
Protected WithEvents form1 As Global.System.Web.UI.HtmlControls.HtmlForm
```

```
'''<summary>  
'''FileUpload1 control.  
'''</summary>  
'''<remarks>  
'''Auto-generated field.  
'''To modify move field declaration from designer file to code-behind file.  
'''</remarks>
```

```
Protected WithEvents FileUpload1 As Global.System.Web.UI.WebControls.FileUpload
```

```
'''<summary>  
'''ButtonUpload control.  
'''</summary>  
'''<remarks>  
'''Auto-generated field.  
'''To modify move field declaration from designer file to code-behind file.  
'''</remarks>
```

```
Protected WithEvents ButtonUpload As Global.System.Web.UI.WebControls.Button
```

```
'''<summary>  
'''ListBoxUploadedImages control.  
'''</summary>  
'''<remarks>  
'''Auto-generated field.  
'''To modify move field declaration from designer file to code-behind file.  
'''</remarks>
```

```
Protected WithEvents ListBoxUploadedImages As Global.System.Web.UI.WebControls.Lis
```

```
'''<summary>  
'''ButtonMakeVideo control.  
'''</summary>  
'''<remarks>  
'''Auto-generated field.  
'''To modify move field declaration from designer file to code-behind file.  
'''</remarks>
```

```
Protected WithEvents ButtonMakeVideo As Global.System.Web.UI.WebControls.Button
```

End Class

```
Imports System.IO
Imports BytescoutImageToVideo
```

```
Partial Public Class _Default
    Inherits Page
```

```
    ' IF YOU SEE TEMPORARY FOLDER ACCESS ERRORS:
```

```
    ' Temporary folder access is required for web application when you use ByteScout S
    ' If you are getting errors related to the access to temporary folder like "Access
```

```
    ' SOLUTION:
```

```
    ' If your IIS Application Pool has "Load User Profile" option enabled the IIS prov
```

```
    ' If you are running Web Application under an impersonated account or IIS_IUSRS gro
```

```
    ' 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 te
- ' - restart your web application and try again

```
Dim uploadPath As String = "\uploads\"
Dim outputPath As String = "\output\"
```

```
Protected Sub ButtonUpload_Click(sender As Object, e As EventArgs) Handles ButtonUp
```

```
    If FileUpload1.HasFile Then
        Dim fileName As String = FileUpload1.FileName
        uploadPath += fileName
        FileUpload1.SaveAs(Server.MapPath(uploadPath))
        ListBoxUploadedImages.Items.Add(FileUpload1.FileName)
    End If
```

```
End Sub
```

```
Protected Sub ButtonMakeVideo_Click(sender As Object, e As EventArgs) Handles Butto
```

```
    ' Create BytescoutImageToVideoLib.ImageToVideo object instance
```

```
Dim converter As New ImageToVideo()
```

```
    ' Activate the component
```

```
converter.RegistrationName = "demo"
```

```
converter.RegistrationKey = "demo"
```

```
Dim outputVideoFile As String = Server.MapPath(outputPath + "result.wmv")
```

```
    ' Add images to converter and set the duration for every slide
```

```
For Each item As ListItem In ListBoxUploadedImages.Items
```

```
    Dim fileName As String = item.Value
```

```
    Dim slide As Slide
```

```
    slide = converter.AddImageFromFileName(Server.MapPath(uploadPath + fileName
```

```
    slide.Duration = 2000 ' 2000ms = 2s
```

```
Next
```

```
    ' Set output video size
```

```
converter.OutputWidth = 640
```

```

converter.OutputHeight = 480

' Set output video file name
converter.OutputVideoFileName = outputVideoFile

' Run the conversion
converter.RunAndWait()

' Release resources
System.Runtime.InteropServices.Marshal.ReleaseComObject(converter)
converter = Nothing

' Show filename of result file
Response.Clear()
Response.ContentType = "text/html"

If File.Exists(outputVideoFile) Then
    Response.Write("Result video file: <b>" + outputVideoFile + "</b>")
Else
    Response.Write("Conversion failed. Error information: " + converter.LastError)
End If

Response.End()

End Sub

End Class

```

SimpleSlideshow.sln

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 15
VisualStudioVersion = 15.0.27703.2026
MinimumVisualStudioVersion = 10.0.40219.1
Project("{F184B08F-C81C-45F6-A57F-5ABD9991F28F}") = "SimpleSlideshow", "SimpleSlideshow
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Any CPU = Debug|Any CPU
        Release|Any CPU = Release|Any CPU
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {5220E41A-A848-4846-886B-B50B9206D646}.Debug|Any CPU.ActiveCfg = Debug
        {5220E41A-A848-4846-886B-B50B9206D646}.Debug|Any CPU.Build.0 = Debug
        {5220E41A-A848-4846-886B-B50B9206D646}.Release|Any CPU.ActiveCfg = Release
        {5220E41A-A848-4846-886B-B50B9206D646}.Release|Any CPU.Build.0 = Release
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution
        HideSolutionNode = FALSE

```

```

EndGlobalSection
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {5741F167-36C9-4307-82FA-FA1CCE935225}
EndGlobalSection
EndGlobal

```

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.

            Visual Basic options:
            Set strict="true" to disallow all data type conversions
            where data loss can occur.
            Set explicit="true" to force declaration of all variables.
        -->
        <compilation debug="true" strict="false" explicit="true" targetFramework="4.0.30319" />
        <pages controlRenderingCompatibilityVersion="3.5" clientIDMode="AutoID" />
        <namespaces>
            <clear/>
            <add namespace="System" />
            <add namespace="System.Collections" />
            <add namespace="System.Collections.Specialized" />
            <add namespace="System.Configuration" />
            <add namespace="System.Text" />
            <add namespace="System.Text.RegularExpressions" />
            <add namespace="System.Web" />
            <add namespace="System.Web.Caching" />
            <add namespace="System.Web.SessionState" />
            <add namespace="System.Web.Security" />
            <add namespace="System.Web.Profile" />
            <add namespace="System.Web.UI" />
            <add namespace="System.Web.UI.WebControls" />
            <add namespace="System.Web.UI.WebControls.WebParts" />
            <add namespace="System.Web.UI.HtmlControls" />
        </namespaces>
    </pages>
    <!--
        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=FzPgeGFL8YA>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Image To Video SDK Home Page](#)
[Explore ByteScout Image To Video SDK Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Image To Video SDK 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)

