How to capture screen video and add time stamp in VB.NET with ByteScout Screen Capturing SDK

How to capture screen video and add time stamp in VB.NET

Learn how to capture screen video and add time stamp in VB.NET with this source code sample. ByteScout Screen Capturing SDK can capture screen video and add time stamp. It can be used from VB.NET. ByteScout Screen Capturing SDK is the tool for developers who want to add screen capturing in their application. Can record screen into video and into single screenshots. Output formats are WMV, AVI, WebM for video and PNG for screenshots. You can adjust output video size, quality, resolution, framerate, video and audio codecs. Includes special privacy features for blacking out sensitive information on screen. Can also capture video from web camera, can add overlays with text or images.

This code snippet below for ByteScout Screen Capturing SDK works best when you need to quickly capture screen video and add time stamp in your VB.NET application. In your VB.NET project or application you may simply copy & paste the code and then run your app! This basic programming language sample code for VB.NET will do the whole work for you to capture screen video and add time stamp.

ByteScout Screen Capturing SDK free trial version is available on our website. VB.NET and other programming languages are supported.

VB.NET - Module1.vb

```
Imports System.Threading
Imports System.Diagnostics

Imports BytescoutScreenCapturingLib ' import bytescout screen capturer activex object


' NOTE: if you are getting error like "invalid image" related to loading the SDK's
dll then
' try to do the following:
' 1) remove the reference to the SDK by View - Solution Explorer
' then click on References, select Bytescout... reference name and right-click it and
select Remove
' 2) To re-add click on the menu: Project - Add Reference
' 3) In "Add Reference" dialog switch to "COM" tab and find Bytescout...
' 4) Select it and click "Add"
' 5) Recompile the application
' Note: if you need to run on both x64 and x86 then please make sure you have set
"Embed Interop Types" to True for this reference


Module Module1
```

```
    Sub Main()
        ' create capturer class
        Dim capturer As New Capturer()
        ' set capturing area to the region type (to capture from given region on the
screen)
        capturer.CapturingType = CaptureAreaType.catScreen
        ' output video filename
        capturer.OutputFileName = "EntireScreenCaptured.wmv" ' set output video
filename to .WMV or .AVI file


        ' set width and height of output video
        capturer.OutputWidth = 640
        capturer.OutputHeight = 480

        ' // WMV and WEBM output use WMVVideoBitrate property to control output video
bitrate
        ' // so try to increase it by x2 or x3 times if you think the output video
are you are getting is laggy
        ' capturer.WMVVideoBitrate = capturer.WMVVideoBitrate * 2


        ' set the text overlay with autochanging time stamp macros to indicate
playing time and current date time
        capturer.OverlayingRedTextCaption = "Recording: {RUNNINGMIN}:{RUNNINGSEC}:
{RUNNINGMSEC} on {CURRENTYEAR}-{CURRENTMONTH}-{CURRENTDAY} at {CURRENTHOUR}:
{CURRENTMIN}:{CURRENTSEC}:{CURRENTMSEC}"


        ' uncomment to set Bytescout Lossless Video format output video compression
method
        ' do not forget to set file to .avi format if you use Video Codec Name
        ' capturer.CurrentVideoCodecName = "Bytescout Lossless"

        ' uncomment to enable recording of semitransparent or layered windows
(Warning: may cause mouse cursor flickering)
        ' capturer.CaptureTransparentControls = true

        ' start capturing
        capturer.Run()

        ' IMPORTANT: if you want to check for some code if need to stop the recording
then make sure you are
        ' using Thread.Sleep(1) inside the checking loop, so you have the loop like
        ' Do
        ' Thread.Sleep(1)
        ' While StopButtonNotClicked


        Console.WriteLine("Capturing entire screen for 25 seconds...")

        ' wait for 25 seconds
        Thread.Sleep(25000)

        ' stop capturing and flush AVI video file into the disk
        capturer.Stop()

        ' Release(Resources)
        System.Runtime.InteropServices.Marshal.ReleaseComObject(capturer)
        capturer = Nothing
        Console.WriteLine("Done.")

        ' open the output video
        Process.Start("EntireScreenCaptured.wmv")

    End Sub

End Module
```

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Screen Capturing SDK

Explore documentation

Visit www.ByteScout.com

or

Get Your Free API Key for www.PDF.co Web API