How to set low FPS to get min file size for screen video in VB.NET with ByteScout Screen Capturing SDK

Tutorial on how to set low FPS to get min file size for screen video in VB.NET

Source code documentation samples provide quick and easy way to add a required functionality into your application. Want to set low FPS to get min file size for screen video in your VB.NET app? ByteScout Screen Capturing SDK is designed for it. ByteScout Screen Capturing SDK is the screen video recording SDK helps in quick implementation of screen video recording. WMV, AVI, WebM output options are available with adjustable quality, video size, framerate and video and audio codec. Includes special features like live multiple blacking out of selected areas, recording from web cam as main source and as overlay, optional watermarks for output video.

You will save a lot of time on writing and testing code as you may just take the VB.NET code from ByteScout Screen Capturing SDK for set low FPS to get min file size for screen video below and use it in your application. Follow the instructions from the scratch to work and copy the VB.NET code. Enjoy writing a code with ready-to-use sample codes in VB.NET.

Trial version of ByteScout Screen Capturing SDK is available for free. Source code samples are included to help you with your VB.NET app.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Screen Capturing SDK

Explore API Documentation

Get Free Training for ByteScout Screen Capturing SDK

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

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
' 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
' 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 "Embe


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 s
        capturer.CapturingType = CaptureAreaType.catScreen
        ' output video filename
        capturer.OutputFileName = "LowFPS.wmv" ' set output video filename to .WMV or


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

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


        ' set WMV video codec to Windows Media Video 9 Screen that gives best quality/
        capturer.CurrentWMVVideoCodecName = "Windows Media Video 9 Screen"
        ' disable audio so we will have a video only (and the lowest wmv file size as
        capturer.AudioEnabled = False
        ' set FPS to 0.5 fps (1 frame per every 2 seconds)
        ' you may also want to set to 1 fps (1 frame per 1 second or higher)
        capturer.FPS = 0.50

        ' set the text overlay with autochanging time stamp macros to indicate playing
        capturer.OverlayingRedTextCaption = "Recording: {RUNNINGMIN}:{RUNNINGSEC}:{RUN

        ' uncomment to enable recording of semitransparent or layered windows (Warning
        ' capturer.CaptureTransparentControls = true


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

        ' start capturing.
        capturer.Run()

        ' IMPORTANT: if you want to check for some code if need to stop the recording
        ' 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 5 seconds...")

        ' wait for 5 seconds
        Thread.Sleep(5000)

        ' 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("LowFPS.wmv")

    End Sub

End Module
```

VIDEO

https://www.youtube.com/watch?v=fujkvtWUVCw

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Screen Capturing SDK Home Page
Explore ByteScout Screen Capturing SDK Documentation