How to use low FPS and min file size for video in C# and ByteScout Screen Capturing SDK

Write code in C# to use low FPS and min file size for video with this step-by-step tutorial

ByteScout tutorials are designed to explain the code for both C# beginners and advanced programmers. ByteScout Screen Capturing SDK can use low FPS and min file size for video. It can be used from C#. 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.

You will save a lot of time on writing and testing code as you may just take the C# code from ByteScout Screen Capturing SDK for use low FPS and min file size for video below and use it in your application. In your C# project or application you may simply copy & paste the code and then run your app! This basic programming language sample code for C# will do the whole work for you to use low FPS and min file size for video.

Our website provides trial version of ByteScout Screen Capturing SDK for free. It also includes documentation and source code samples.

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:

Program.cs

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;
using System.Diagnostics;
using System.Drawing;
using BytescoutScreenCapturingLib; // import bytescout screen capturing activex object

// NOTE: if you are getting error like "invalid image" related to loading the SDK's dl
// 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 "Emb

namespace SimpleCaptureCSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            Capturer capturer = new Capturer(); // create new screen capturer object


            capturer.CapturingType = CaptureAreaType.catScreen; // set capturing area t

            capturer.OutputFileName = "LowFPS.wmv"; // set output video filename to .WM

        // 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 = 0.50f;

            // 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 ar
            // capturer.WMVVideoBitrate = capturer.WMVVideoBitrate * 2;



            // set output video width and height
            capturer.OutputWidth = 640;
            capturer.OutputHeight = 480;

            // set the text overlay with autochanging time stamp macros to indicate pl
            capturer.OverlayingRedTextCaption = "Recording: {RUNNINGMIN}:{RUNNINGSEC}:
```

```
            // uncomment to enable recording of semitransparent or layered windows (Wa
            // capturer.CaptureTransparentControls = true;

            // set border around captured area if we are not capturing entire screen
            if (
                capturer.CapturingType != CaptureAreaType.catScreen &&
                capturer.CapturingType != CaptureAreaType.catWebcamFullScreen
                )
            {
                // set border style
                capturer.CaptureAreaBorderType = CaptureAreaBorderType.cabtDashed;
                capturer.CaptureAreaBorderColor = (uint)ColorTranslator.ToOle(Color.Re
            }


            capturer.Run(); // run screen video capturing

        // 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...");

            Thread.Sleep(5000); // wait for 5 seconds

            capturer.Stop(); // stop video capturing

            // Release resources
            System.Runtime.InteropServices.Marshal.ReleaseComObject(capturer);
            capturer = null;


            Console.WriteLine("Done");


            Process.Start("LowFPS.wmv");
        }
    }
}
```

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
Explore Samples
Sign Up for ByteScout Screen Capturing SDK 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