

blackout sensitive info on screen video in VB.NET and ByteScout Screen Capturing SDK

blackout sensitive info on screen video in VB.NET

We've created and updating regularly our sample code library so you may quickly learn blackout sensitive info on screen video and the step-by-step process in VB.NET. ByteScout Screen Capturing SDK helps with blackout sensitive info on screen video in VB.NET. ByteScout Screen Capturing SDK is the SDK for developers for quick implementation of screen video recording. The SDK records screen into video or into a series of screenshots. Can also record audio. Saves video into AVI, WMV and Google's WebM. Output video quality, size, resolution or framerate can be adjusted easily. Provides additional tools for privacy features like blacking out on screen areas with sensitive information on screen right during recording. Supports web camera as input and can add instant text and images into video output.

This rich sample source code in VB.NET for ByteScout Screen Capturing SDK includes the number of functions and options you should do calling the API to implement blackout sensitive info on screen video. In order to implement this functionality, you should copy and paste code below into your app using code editor. Then compile and run your application. Enhanced documentation and tutorials are available along with installed ByteScout Screen Capturing SDK if you'd like to dive deeper into the topic and the details of the API.

Visit our website provides for free trial version of ByteScout Screen Capturing SDK. Free trial includes lots of source code samples to help you with your VB.NET project.

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](http://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 s
' 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 = "EntireScreenCaptured.wmv" ' set output video filename

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

        ' 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"

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

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

        ' 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);

' 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 25 seconds...")

' wait for 1 second (1000 msec)
Thread.Sleep(1000)

Console.WriteLine("Adding blackout regions after 1 sec of recording...")

capturer.BlackoutAddArea(20, 100, 500, 80)
capturer.BlackoutAddArea(10, 200, 550, 80)
capturer.BlackoutAddArea(20, 400, 550, 120)

' wait for 15 seconds (15000 msec)
Thread.Sleep(15000)

Console.WriteLine("Remove blackout regions...")

' reset all blackouts
capturer.BlackoutReset()

' wait for 9 seconds more
Thread.Sleep(9000)

' 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

```

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](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com