take screenshots during recording in VB.NET with ByteScout Screen Capturing SDK

How To: tutorial on take screenshots during recording in VB.NET

These source code samples are listed and grouped by their programming language and functions they use. ByteScout Screen Capturing SDK was made to help with take screenshots during recording in 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.

VB.NET code snippet like this for ByteScout Screen Capturing SDK works best when you need to quickly implement take screenshots during recording in your VB.NET application. To do take screenshots during recording in your VB.NET project or application you may simply copy & paste the code and then run your app! You can use these VB.NET sample examples in one or many applications.

Trial version can be obtained from our website for free. It includes this and other source code samples for VB.NET.

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:

Form1.Designer.vb

```vb
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Form1
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
                Me.buttonScreenshot = New System.Windows.Forms.Button()
                Me.buttonStop = New System.Windows.Forms.Button()
                Me.buttonStart = New System.Windows.Forms.Button()
                Me.SuspendLayout()
                '
                'buttonScreenshot
                '
                Me.buttonScreenshot.Enabled = False
                Me.buttonScreenshot.Location = New System.Drawing.Point(108, 41)
                Me.buttonScreenshot.Name = "buttonScreenshot"
                Me.buttonScreenshot.Size = New System.Drawing.Size(100, 23)
                Me.buttonScreenshot.TabIndex = 5
                Me.buttonScreenshot.Text = "Take Screenshot"
                Me.buttonScreenshot.UseVisualStyleBackColor = True
                '
                'buttonStop
                '
                Me.buttonStop.Enabled = False
                Me.buttonStop.Location = New System.Drawing.Point(27, 58)
                Me.buttonStop.Name = "buttonStop"
                Me.buttonStop.Size = New System.Drawing.Size(75, 23)
                Me.buttonStop.TabIndex = 4
                Me.buttonStop.Text = "Stop"
                Me.buttonStop.UseVisualStyleBackColor = True
                '
                'buttonStart
                '
                Me.buttonStart.Location = New System.Drawing.Point(27, 29)
                Me.buttonStart.Name = "buttonStart"
```

```vb
            Me.buttonStart.Size = New System.Drawing.Size(75, 23)
            Me.buttonStart.TabIndex = 3
            Me.buttonStart.Text = "Start"
            Me.buttonStart.UseVisualStyleBackColor = True
            '
            'Form1
            '
            Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
            Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
            Me.ClientSize = New System.Drawing.Size(234, 111)
            Me.Controls.Add(Me.buttonScreenshot)
            Me.Controls.Add(Me.buttonStop)
            Me.Controls.Add(Me.buttonStart)
            Me.Name = "Form1"
            Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
            Me.Text = "Form1"
            Me.ResumeLayout(False)

    End Sub
    Private WithEvents buttonScreenshot As System.Windows.Forms.Button
    Private WithEvents buttonStop As System.Windows.Forms.Button
    Private WithEvents buttonStart As System.Windows.Forms.Button

End Class
```

Form1.vb

```vb
Imports BytescoutScreenCapturingLib

Public Class Form1

        ' NOTE: if you are getting error like "invalid image" related to loading the SD
        ' 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
        ' 2) To re-add click on the menu: Project - Add Reference
        ' 3) In "Add Reference" dialog switch to "COM" tab and find "BytescoutScreenCap
        ' 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 s

    Dim _capturer As Capturer = Nothing

    Dim _screencastCount As Integer = 0
    Dim _screenshotCount As Integer = 0

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

            ' Create and setup Capturer object:
```

```vb
        _capturer = New Capturer()

        ' Capture the full screen
        _capturer.CapturingType = CaptureAreaType.catScreen

        ' Set movie width and height to the current monitor dimensions
        _capturer.OutputWidth = _capturer.MonitorWidth
        _capturer.OutputHeight = _capturer.MonitorHeight

        ' Setup screenshots
        _capturer.ScreenshotImageFormat = ScreenshotImageType.ssitPng ' Save in
        _capturer.ScreenshotOutputFolder = "\." ' Save to the current folder
        _capturer.ScreenshotSavingType = ScreenshotSavingType.ssstManual ' Take

    End Sub

    Private Sub buttonStart_Click(sender As Object, e As EventArgs) Handles buttonS

        _screencastCount = _screencastCount + 1
        Dim fileName As String = String.Format("screencast{0}.wmv", _screencas

        ' Start screen recording
        _capturer.OutputFileName = fileName
        _capturer.Run()

        buttonStart.Enabled = False
        buttonStop.Enabled = True
        buttonScreenshot.Enabled = True

    End Sub

    Private Sub buttonStop_Click(sender As Object, e As EventArgs) Handles buttonSt

        ' Stop recording
        _capturer.Stop()

        buttonStart.Enabled = True
        buttonStop.Enabled = False
        buttonScreenshot.Enabled = False

        MessageBox.Show("Screencast saved to " + _capturer.OutputFileName, Prod

    End Sub

    Private Sub buttonScreenshot_Click(sender As Object, e As EventArgs) Handles bu

        _screenshotCount = _screenshotCount + 1
        Dim fileName As String = String.Format("screenshot{0}.png", _screensho

        ' Take screenshot during the recording
        _capturer.TakeScreenshot(fileName)

        MessageBox.Show("Screenshot saved to " + fileName, ProductName)

    End Sub

End Class
```

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