

take screenshots on timer in VBScript and VB6 with ByteScout Screen Capturing SDK

Make take screenshots on timer in VBScript and VB6

:

Tutorial on how to do take screenshots on timer in VBScript and VB6

Here you may find thousands pre-made source code pieces for easy implementation in your own programming VBScript and VB6 projects. ByteScout Screen Capturing SDK was made to help with take screenshots on timer in VBScript and VB6. 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 rich sample source code in VBScript and VB6 for ByteScout Screen Capturing SDK includes the number of functions and options you should do calling the API to implement take screenshots on timer. Follow the instruction from the scratch to work and copy and paste code for VBScript and VB6 into your editor. Enjoy writing a code with ready-to-use sample VBScript and VB6 codes to add take screenshots on timer functions using ByteScout Screen Capturing SDK in VBScript and VB6.

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 VBScript and VB6 project.

VBScript and VB6 - CaptureScreenAndTakeScreenshotsOnTimer.vbs

```
' create video capturer activex object
Set capturer = CreateObject("BytescoutScreenCapturing.Capturer")

' set output video file name to .WMV or .AVI file
capturer.OutputFileName = "EntireScreenCaptured.wmv"

' set capturing type to the caScreen =3 to capture the entire screen
capturer.CapturingType = 3

' output video width
capturer.OutputWidth = 640

' output video height
capturer.OutputHeight = 480

' 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

' set the screenshot saving mode type: 0=none (default), 1=onmouseclick, 2=ontimer,
3>manual (using TakeScreenshot() method only)
capturer.ScreenshotSavingType = 2
' set output folder for the screenshots

set fso = CreateObject("Scripting.FileSystemObject")
' directory in which this script is currently running
CurrentDirectory = fso.GetAbsolutePathName(".")

capturer.ScreenshotOutputFolder = CurrentDirectory ' set output folder to the current
one
' set image format for the screenshots (png = 0, bmp = 1)
capturer.ScreenshotImageFormat = 0
' set timer period for taking screenshots, in ms
capturer.ScreenshotTimerPeriod = 1000 ' set the timer for 1 second

MsgBox "This script will record 640x480 video from entire screen for 15 seconds.
Output video will go to EntireScreenCaptured.avi"

' run video 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

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

' you may also make manual screenshots
capturer.TakeScreenshot "screenshot1.png"

' stop capturing
capturer.Stop()

' destroy Capturer object so the video will be saved into the disk
Set capturer = Nothing
```

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

or

[Get Your Free API Key for www.PDF.co Web API](#)