How to capture video from entire screen as WEBM in C++ (managed) using ByteScout Screen Capturing SDK

Tutorial on how to capture video from entire screen as WEBM in C++ (managed)

The sample source codes on this page shows how to capture video from entire screen as WEBM in C++ (managed). What is ByteScout Screen Capturing SDK? It 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. It can help you to capture video from entire screen as WEBM in your C++ (managed) application.

C++ (managed) code samples for C++ (managed) developers help to speed up coding of your application when using ByteScout Screen Capturing SDK. In order to implement the functionality, you should copy and paste this code for C++ (managed) below into your code editor with your app, compile and run your application. Further enhancement of the code will make it more vigorous.

Download free trial version of ByteScout Screen Capturing SDK from our website with this and other source code samples for C++ (managed).

C++ (managed) - AssemblyInfo.cpp

```
#include "stdafx.h"

using namespace System;
using namespace System::Reflection;
using namespace System::Runtime::CompilerServices;
using namespace System::Runtime::InteropServices;
using namespace System::Security::Permissions;

//
// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
//
[assembly:AssemblyTitleAttribute("CaptureFromEntireScreen")];
[assembly:AssemblyDescriptionAttribute("")];
[assembly:AssemblyConfigurationAttribute("")];
[assembly:AssemblyCompanyAttribute("")];
[assembly:AssemblyProductAttribute("CaptureFromEntireScreen")];
[assembly:AssemblyCopyrightAttribute("Copyright (c)  2011")];
[assembly:AssemblyTrademarkAttribute("")];
[assembly:AssemblyCultureAttribute("")];
```

```
//
// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
// You can specify all the value or you can default the Revision and Build Numbers
// by using the '*' as shown below:

[assembly:AssemblyVersionAttribute("1.0.*")];

[assembly:ComVisible(false)];

[assembly:CLSCompliantAttribute(true)];

[assembly:SecurityPermission(SecurityAction::RequestMinimum, UnmanagedCode = true)];
```

C++ (managed) - CaptureFromEntireScreen.cpp

```
// CaptureFromEntireScreen.cpp : main project file.

#include "stdafx.h"

using namespace System;
using namespace System::Threading;
using namespace System::Diagnostics;

using namespace BytescoutScreenCapturingLib;


int main(array ^args)
{
        // Create Capturer instance
    Capturer ^capturer = gcnew Capturer();

        capturer->RegistrationName = "demo";
        capturer->RegistrationKey = "demo";

        // Set capturing type
        capturer->CapturingType = CaptureAreaType::catScreen;

        // Set output video width and height
        capturer->OutputWidth = 640;
        capturer->OutputHeight = 480;

            // WMV and WEBM output use WMVVideoBitrate property to control output
video bitrate
            // so try to increase it by x2 or x3 times if you think the output video
```

```
are you are getting is laggy
            // capturer->put_WMVVideoBitrate(capturer->WMVVideoBitrate * 2);


        // uncomment to enable recording of semitransparent or layered windows
(Warning: may cause mouse cursor flickering)
        // capturer->CaptureTransparentControls = true;


        // Set output file name
        capturer->OutputFileName = "Output.webm";

        // Start 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);


        Console::WriteLine("Capture the desktop for 5s...");

        // Wait for 5 seconds
        Thread::Sleep(5000);

        // Stop capturing
        capturer->Stop();

    // Release resources
    System::Runtime::InteropServices::Marshal::ReleaseComObject(capturer);
    //capturer = NULL;


        Console::WriteLine("Done.");

        // Open the capture video in default associated application
        Process::Start("Output.webm");

        return 0;
}
```

C++ (managed) - resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by app.rc
```

C++ (managed) - stdafx.cpp

```
// stdafx.cpp : source file that includes just the standard includes
// CaptureFromEntireScreen.pch will be the pre-compiled header
// stdafx.obj will contain the pre-compiled type information

#include "stdafx.h"
```

C++ (managed) - stdafx.h

```
// stdafx.h : include file for standard system include files,
// or project specific include files that are used frequently, but
// are changed infrequently
//

#pragma once

// TODO: reference additional headers your program requires here
```

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]

or

Get Your Free API Key for www.PDF.co Web API