www.bytescout.com

How to record screen video and add time stamp in C++ (managed) and ByteScout Screen Capturing SDK

This code in C++ (managed) shows how to record screen video and add time stamp with this how to tutorial

The sample shows steps and algorithm of how to record screen video and add time stamp and how to make it work in your C++ (managed) application. ByteScout Screen Capturing SDK: the screen video recording SDK helps in quick implementation of screen video recording. WMV, AVI, WebM output options are available with adjustable quality, video size, framerate and video and audio codec. Includes special features like live multiple blacking out of selected areas, recording from web cam as main source and as overlay, optional watermarks for output video. It can record screen video and add time stamp in C++ (managed).

The SDK samples like this one below explain how to quickly make your application do record screen video and add time stamp in C++ (managed) with the help of ByteScout Screen Capturing SDK. In your C++ (managed) project or application you may simply copy & paste the code and then run your app! Test C++ (managed) sample code examples whether they respond your needs and requirements for the project.

ByteScout Screen Capturing SDK free trial version is available on our website. C++ (managed) and other programming languages are supported.

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:

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)];
```

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<System::String ^> ^args)
{
        // Create Capturer instance
    Capturer ^capturer = gcnew Capturer();

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

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

        // show recording time stamp
        capturer->OverlayingRedTextCaption = "Recording: {RUNNINGMIN}:{RUNNINGSEC}:{RU

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

            // WMV and WEBM output use WMVVideoBitrate property to control output vide
            // so try to increase it by x2 or x3 times if you think the output video a
            // capturer->put_WMVVideoBitrate(capturer->WMVVideoBitrate * 2);


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

        // uncomment to enable recording of semitransparent or layered windows (Warnin
        // capturer->CaptureTransparentControls = true;


        // 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);


        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.wmv");

        return 0;
}
```

resource.h

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

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"
```

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
```

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