www.bytescout.com

How to capture video from given window in C++ (unmanaged) using ByteScout Screen Capturing SDK

Write code in C++ (unmanaged) to capture video from given window with this step-by-step tutorial

Capture video from given window is easy to implement in C++ (unmanaged) if you use these source codes below. 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 and you can use it to capture video from given window with C++ (unmanaged).

The SDK samples like this one below explain how to quickly make your application do capture video from given window in C++ (unmanaged) with the help of ByteScout Screen Capturing SDK. In your C++ (unmanaged) project or application you may simply copy & paste the code and then run your app! Code testing will allow the function to be tested and work properly with your data.

Free trial version of ByteScout Screen Capturing SDK is available for download from our website. Get it to try other source code samples for C++ (unmanaged).

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:

CaptureFromGivenWindow.cpp

```
// CaptureFromEntireScreen.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#import "BytescoutScreenCapturing.dll"

using namespace BytescoutScreenCapturingLib;
using namespace std;


void usage(ICapturer* capturer);
void setParams(int argc, _TCHAR* argv[], ICapturer* capturer);


int _tmain(int argc, _TCHAR* argv[])
{
	::CoInitialize(0);

	// Create Capturer instance

	CLSID clsid_ScreenCapturer;
	CLSIDFromProgID(OLESTR("BytescoutScreenCapturing.Capturer"), &clsid_ScreenCaptu

	ICapturer* capturer = NULL;
	::CoCreateInstance(clsid_ScreenCapturer, NULL, CLSCTX_ALL, __uuidof(ICapturer)

	if (!capturer)
	{
		_ftprintf(stdout, _T("Screen Capturer is not installed properly."));
		::CoUninitialize();
		return 1;
	}

	capturer->put_RegistrationName(_T("demo"));
	capturer->put_RegistrationKey(_T("demo"));

	// Set capturing type
	capturer->put_CapturingType(catWindow);

	// Specify window to capture by its title.
	// Capturer will take the first window containing "Internet Explorer" in its t
	capturer->put_WindowToCapture(_T("Internet Explorer"));

	// You can also capture only a part of the window by specifying capturing recto
	//capturer->put_CaptureRectLeft(25);
	//capturer->put_CaptureRectTop(25);
	//capturer->put_CaptureRectWidth(320);
	//capturer->put_CaptureRectHeight(240);

	// Set output video width and height
	capturer->put_OutputWidth(640);
	capturer->put_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);


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


    // set border style
    ?apturer.CaptureAreaBorderType = cabtDashed;

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

    // Start capturing
    HRESULT hr = 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


    if (FAILED(hr))
    {
            // Error handling
            CComBSTR s;
            capturer->get_LastError(&s);
            _ftprintf(stdout, _T("Capture failed: %s\n"), CString(s));
    }
    else
    {
            _tprintf(_T("Starting capture - Hit a key to stop ...\n"));

            int i = 0;
            TCHAR *spin = _T("|/-\\");

            // Show some progress
            while (!_kbhit())
            {
                    _tprintf(_T("\rEncoding %c"), spin[i++]);
                    i %= 4;
                    Sleep(50);
            }

            // Stop after key press
            capturer->Stop();

            _tprintf(_T("\nDone."));
            getchar();
    }

    // Release Capturer
    capturer->Release();
    capturer = NULL;

    ::CoUninitialize();
```

```
        return 0;
}
```

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"

// TODO: reference any additional headers you need in STDAFX.H
// and not in this file
```

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

#ifndef _WIN32_WINNT            // Allow use of features specific to Windows XP or late
#define _WIN32_WINNT 0x0501     // Change this to the appropriate value to target other
#endif

#include <stdio.h>
#include <tchar.h>

#include <atlbase.h>
#include <atlstr.h>
#include <conio.h>
```

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