

How to capture video from entire screen in C++ (unmanaged) and ByteScout Screen Capturing SDK

This tutorial will show how to capture video from entire screen in C++ (unmanaged)

Learn how to capture video from entire screen in C++ (unmanaged) with this source code sample. ByteScout Screen Capturing SDK: 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 capture video from entire screen in C++ (unmanaged).

Fast application programming interfaces of ByteScout Screen Capturing SDK for C++ (unmanaged) plus the instruction and the code below will help you quickly learn how to capture video from entire screen. This C++ (unmanaged) sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! This basic programming language sample code for C++ (unmanaged) will do the whole work for you to capture video from entire screen.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with source code samples.

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

Source Code Files:

CaptureFromEntireScreen.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_ScreenCapturer);

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

    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(catScreen);

    // Set output video width and height
    capturer->put_OutputWidth(640);
    capturer->put_OutputHeight(480);

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

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

    // uncomment to enable recording of semitransparent or layered windows (Warning: may be slow)
```

```

// capturer->CaptureTransparentControls = true;

// 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 : 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

#ifdef _WIN32_WINNT           // Allow use of features specific to Windows XP or later
#define _WIN32_WINNT 0x0501 // Change this to the appropriate value to target other versions of Windows
#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](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com