

## How to record screen video from given window in C++ (managed) with ByteScout Screen Capturing SDK

### How to record screen video from given window in C++ (managed)

The sample source code below will teach you how to record screen video from given window in C++ (managed). Want to record screen video from given window in your C++ (managed) app? ByteScout Screen Capturing SDK is designed for it. ByteScout Screen Capturing SDK is 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.

You will save a lot of time on writing and testing code as you may just take the C++ (managed) code from ByteScout Screen Capturing SDK for record screen video from given window below and use it in your application. Follow the instructions from the scratch to work and copy the C++ (managed) code. Detailed tutorials and documentation are available along with installed ByteScout Screen Capturing SDK if you'd like to dive deeper into the topic and the details of the API.

Trial version of ByteScout Screen Capturing SDK can be downloaded for free from our website. It also includes source code samples for C++ (managed) and other programming languages.

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:

## 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: AssemblyTitle("CaptureFromGivenWindow")];
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfigurationAttribute("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("CaptureFromGivenWindow")];
[assembly: AssemblyCopyright("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: AssemblyVersion("1.0.*")];

[assembly: ComVisible(false)];

[assembly: CLSCompliant(true)];

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

## CaptureFromGivenWindow.cpp

```

// CaptureFromGivenWindow.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 = gnew Capturer();

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

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

    // Specify window to capture by its title.
    // Capturer will take the first window containing "Internet Explorer" in its title
    capturer->WindowToCapture = "Internet Explorer";

    // You can also capture only a part of the window by specifying capturing rectangle
    //capturer->CaptureRectLeft = 25;
    //capturer->CaptureRectTop = 25;
    //capturer->CaptureRectWidth = 320;
    //capturer->CaptureRectHeight = 240;

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

    // uncomment to enable recording of semitransparent or layered windows (Warning: may be slow)
    // capturer->CaptureTransparentControls = true;

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

    // set border style
    capturer.CaptureAreaBorderType = CaptureAreaBorderType::cabtDashed;

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

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

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

