

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

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

We made thousands of pre-made source code pieces for easy implementation in your own programming projects. 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 record screen video from entire screen 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. Follow the instructions from the scratch to work and copy the C++ (managed) code. Code testing will allow the function to be tested and work properly with your data.

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:

## 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("CaptureFromEntireScreen")];
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfigurationAttribute("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("CaptureFromEntireScreen")];
[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)];
```

## 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;

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

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

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

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

[www.bytescout.com](http://www.bytescout.com)