

How to read barcode from memory with barcode reader sdk in C++ and ByteScout Data Extraction Suite

Step-by-step tutorial on how to read barcode from memory with barcode reader sdk in C++

Read barcode from memory with barcode reader sdk is simple to apply in C++ if you use these source codes below. ByteScout Data Extraction Suite is the set that includes 3 SDK products for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK. It can read barcode from memory with barcode reader sdk in C++.

This prolific sample source code in C++ for ByteScout Data Extraction Suite contains various functions and other necessary options you should do calling the API to read barcode from memory with barcode reader sdk. Follow the instructions from scratch to work and copy the C++ code. This basic programming language sample code for C++ will do the whole work for you to read barcode from memory with barcode reader sdk.

The trial version of ByteScout Data Extraction Suite can be downloaded for free from our website. It also includes source code samples for C++ and other programming languages.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout Data Extraction Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Data Extraction Suite](#)

[Get Free API key for Web API](#)

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

Source Code Files:

ReadFromMemory.cpp

```

// ReadFromMemory.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "atlbase.h"

#import "c:\\Program Files\\Bytescout BarCode Reader SDK\\net4.00\\Bytescout.BarCodeRe
using namespace Bytescout_BarCodeReader;

int _tmain(int argc, _TCHAR* argv[])
{
    // Initialize COM.
    HRESULT hr = CoInitialize(NULL);

    // Create the interface pointer.
    IReaderPtr pIReader(__uuidof(Reader));

    // set the registration name and key
    BSTR regname = ::SysAllocString(L"DEMO");
    pIReader->put_RegistrationName(regname);
    SysFreeString(regname);
    BSTR regkey = ::SysAllocString(L"DEMO");
    pIReader->put_RegistrationKey(regkey);
    SysFreeString(regkey);

    // Set barcode type to find
    _BarcodeTypeSelectorPtr pBarcodeTypesToFind;
    pIReader->get_BarcodeTypesToFind(&pBarcodeTypesToFind);
    pBarcodeTypesToFind->put_GS1DataBarExpanded(VARIANT_TRUE);

    // Get full path of sample barcode image file
    WCHAR file[MAX_PATH];
    ::GetFullPathName(L"GS1DataBarExpanded.png", MAX_PATH, file, NULL);

    // Load file content to byte array (for demonstration purpose)
    HANDLE hFile = CreateFile(file, GENERIC_READ, FILE_SHARE_READ, NULL, OPEN_EXISTING);
    DWORD fileSize = GetFileSize(hFile, NULL);
    byte* pBuffer = new byte[fileSize];
    DWORD numberOfBytesRead;
    ReadFile(hFile, pBuffer, fileSize, &numberOfBytesRead, NULL);
    CloseHandle(hFile);

    // Read barcode from memory
    IStream* stream = SHCreateMemStream(pBuffer, fileSize);
    hr = pIReader->ReadFromStream(stream);

    stream->Release();
    delete[] pBuffer;

    // Get found barcode count
    long count;
    pIReader->get_FoundCount(&count);

    // Get found barcode information
    for (int i = 0; i < count; i++)
    {
        SymbologyType type;
        hr = pIReader->GetFoundBarcodeType(i, &type);
    }
}

```

```

        wprintf(L"Barcode type: %d\n", type);

        float confidence;
        hr = pIReader->GetFoundBarcodeConfidence(i, &confidence);
        wprintf(L"Barcode confidence: %f\n", confidence);

        BSTR bstrValue;
        hr = pIReader->GetFoundBarcodeValue(i, &bstrValue);
        wprintf(L"Barcode value: %s\n", bstrValue);
        ::SysFreeString(bstrValue);

        wprintf(L"\n");
    }

    if (count == 0)
    {
        wprintf(L"no barcodes found");
    }

    // Uninitialize COM.
    CoUninitialize();

    wprintf(L"\nHit key to continue\n");
    getchar();

    return 0;
}

```

ReadFromMemory.sln

```

Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio 2013
VisualStudioVersion = 12.0.40629.0
MinimumVisualStudioVersion = 10.0.40219.1
Project("{8BC9CEB8-8B4A-11D0-8D11-00A0C91BC942}") = "ReadFromMemory", "ReadFromMemory.v
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Win32 = Debug|Win32
        Release|Win32 = Release|Win32
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {ABBECC62-8779-458A-B8DC-F949363D29AF}.Debug|Win32.ActiveCfg = Debug|Win
        {ABBECC62-8779-458A-B8DC-F949363D29AF}.Debug|Win32.Build.0 = Debug|Win
        {ABBECC62-8779-458A-B8DC-F949363D29AF}.Release|Win32.ActiveCfg = Releas
        {ABBECC62-8779-458A-B8DC-F949363D29AF}.Release|Win32.Build.0 = Release
    EndGlobalSection
    GlobalSection(SolutionProperties) = preSolution

```

```
        HideSolutionNode = FALSE
    EndGlobalSection
EndGlobal
```

stdafx.cpp

```
// stdafx.cpp : source file that includes just the standard includes
// ReadFromMemory.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

#include "targetver.h"

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

// TODO: reference additional headers your program requires here
```

targetver.h

```
#pragma once

// Including SDKDDKVer.h defines the highest available Windows platform.

// If you wish to build your application for a previous Windows platform, include WinS
// set the _WIN32_WINNT macro to the platform you wish to support before including SDK

#include <SDKDDKVer.h>
```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Data Extraction Suite Home Page](#)
[Explore ByteScout Data Extraction Suite Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Data Extraction Suite 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