

How to get barcode to memory buffer with barcode sdk in C++ and ByteScout Premium Suite

How to write a robust code in C++ to get barcode to memory buffer with barcode sdk with this step-by-step tutorial

The code displayed below will guide you to install an C++ app to get barcode to memory buffer with barcode sdk. ByteScout Premium Suite is the bundle that includes twelve SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording. It can be applied to get barcode to memory buffer with barcode sdk using C++.

Want to save time? You will save a lot of time on writing and testing code as you may just take the C++ code from ByteScout Premium Suite for get barcode to memory buffer with barcode sdk below and use it in your application. Simply copy and paste in your C++ project or application you and then run your app! Further improvement of the code will make it more robust.

You can download free trial version of ByteScout Premium Suite from our website to see and try many others source code samples for C++.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Premium Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Premium Suite](#)

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

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

Source Code Files:

```
#include "stdafx.h"
#include <atlsafe.h> // For ATL::CComSafeArray

#import "Bytescout.BarCode.tlb" raw_interfaces_only

using namespace Bytescout_BarCode;

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

    // Create the interface pointer.
    IBarcodePtr pIBarcode(__uuidof(Barcode));

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

    BSTR regkey = ::SysAllocString(L"DEMO");
    pIBarcode->put_RegistrationKey(regkey);
    ::SysFreeString(regkey);

    // Set barcode type (symbology)
    pIBarcode->put_Symbology(SymbologyType_Code128);

    // Set barcode value
    BSTR value = ::SysAllocString(L"Abc123");
    pIBarcode->put_Value(value);
    ::SysFreeString(value);

    // Get barcode image bytes.
    SAFEARRAY* pSafeArray;
    pIBarcode->GetImageBytesPNG(&pSafeArray);

    // Convert SAFEARRAY to byte array
    CComSafeArray<BYTE> safeArray;
    safeArray.Attach(pSafeArray);
    UINT count = safeArray.GetCount();
    BYTE* bytes = new BYTE[count];
    for (ULONG i = 0; i < count; i++)
        bytes[i] = safeArray.GetAt(i);

    // Check result by saving image bytes to file
    FILE* pFile;
    fopen_s(&pFile, "result.png", "wb");
    fwrite(bytes, 1, count, pFile);
    fclose(pFile);

    delete[] bytes;
    pIBarcode->Release();

    // Uninitialize COM.
```

```
    CoUninitialize();  
  
    return 0;  
}
```

BarcodeGenerationExample.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}") = "BarcodeGenerationExample", "BarcodeGenerationExample.sln", "{...}"  
EndProject  
Global  
    GlobalSection(SolutionConfigurationPlatforms) = preSolution  
        Debug|Win32 = Debug|Win32  
        Release|Win32 = Release|Win32  
    EndGlobalSection  
    GlobalSection(ProjectConfigurationPlatforms) = postSolution  
        {7197580A-6CCC-4581-BD61-9028B6A0578E}.Debug|Win32.ActiveCfg = Debug|Win32  
        {7197580A-6CCC-4581-BD61-9028B6A0578E}.Debug|Win32.Build.0 = Debug|Win32  
        {7197580A-6CCC-4581-BD61-9028B6A0578E}.Release|Win32.ActiveCfg = Release|Win32  
        {7197580A-6CCC-4581-BD61-9028B6A0578E}.Release|Win32.Build.0 = Release|Win32  
    EndGlobalSection  
    GlobalSection(SolutionProperties) = preSolution  
        HideSolutionNode = FALSE  
    EndGlobalSection  
EndGlobal
```

stdafx.cpp

```
// stdafx.cpp : source file that includes just the standard includes  
// BarcodeGenerationExample.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 Premium Suite Home Page](#)

[Explore ByteScout Premium Suite Documentation](#)

[Explore Samples](#)

[Sign Up for ByteScout Premium 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