How to generate barcode in C++ with ByteScout Barcode SDK

This tutorial will show how to generate barcode in C++

Every ByteScout tool contains example C++ source codes that you can find here or in the folder with installed ByteScout product. ByteScout Barcode SDK: the robost library (Software Development Kit) that is designed for automatic generation of high-quality barcodes for printing, electronic documents and pdf. All popular barcode types are supported from Code 39 and Code 129 to QR Code, UPC, GS1, GS-128, Datamatrix, PDF417, Maxicode and many others. Provides support for full customization of fonts, colors, output and printing sizes. Special tools are included to verify output quality and printing quality. Can add generated barcode into new or existing documents, images and PDF. It can generate barcode in C++.

The SDK samples like this one below explain how to quickly make your application do generate barcode in C++ with the help of ByteScout Barcode SDK. Just copy and paste the code into your C++ application's code and follow the instruction. Detailed tutorials and documentation are available along with installed ByteScout Barcode SDK if you'd like to dive deeper into the topic and the details of the API.

Download free trial version of ByteScout Barcode SDK from our website with this and other source code samples for C++.

C++ - BarcodeGenerationExample.cpp

```
#include "stdafx.h"

#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);

        // Save barcode image
        BSTR fileName = ::SysAllocString(L"result.png");
        pIBarcode->SaveImage(fileName);
        ::SysFreeString(fileName);

        pIBarcode->Release();

        // Uninitialize COM.
        CoUninitialize();

        return 0;
}
```

C++ - 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
```

C++ - 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
#include



// TODO: reference additional headers your program requires here


```

C++ - 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
WinSDKVer.h and
// set the _WIN32_WINNT macro to the platform you wish to support before including
SDKDDKVer.h.

#include


```
