www.bytescout.com

How to generate barcode monochrome bitmap with barcode sdk in C++ with ByteScout Premium Suite

If you want to learn more then this tutorial will show how to generate barcode monochrome bitmap with barcode sdk in C++

The sample source codes on this page shows how to generate barcode monochrome bitmap with barcode sdk in C++. ByteScout Premium Suite can generate barcode monochrome bitmap with barcode sdk. It can be applied from C++. ByteScout Premium Suite is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording.

Want to quickly learn? This fast application programming interfaces of ByteScout Premium Suite for C++ plus the guidelines and the code below will help you quickly learn how to generate barcode monochrome bitmap with barcode 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 generate barcode monochrome bitmap with barcode sdk.

ByteScout provides the free trial version of ByteScout Premium Suite along with the documentation and source code samples.

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

Source Code Files:

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_DataMatrix);

	// Set monochrome output
	pIBarcode->put_ProduceMonochromeImages(VARIANT_TRUE);

	// Set barcode value
	BSTR value = ::SysAllocString(L"Barcode Value 1234567890");
	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;
}
```

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", "Barco
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|W
                {7197580A-6CCC-4581-BD61-9028B6A0578E}.Debug|Win32.Build.0 = Debug|Win
                {7197580A-6CCC-4581-BD61-9028B6A0578E}.Release|Win32.ActiveCfg = Releas
                {7197580A-6CCC-4581-BD61-9028B6A0578E}.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
// 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 WinSD
// set the _WIN32_WINNT macro to the platform you wish to support before including SDKD

#include <SDKDDKVer.h>
```

VIDEO

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