

PDF and OCR (optical character recognition) with PDF extractor SDK in C++ using ByteScout Premium Suite

PDF and OCR (optical character recognition) with PDF extractor SDK in C++

These sample source codes given below will show you how to handle a complex task, for example, PDF and OCR (optical character recognition) with PDF extractor SDK in C++. ByteScout Premium Suite was made to help with PDF and OCR (optical character recognition) with PDF extractor SDK in C++. ByteScout Premium Suite is the bundle that includes twelve SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording.

Want to speed up the application development? Then this C++, code samples for C++, developers help to speed up the application development and writing a code when using ByteScout Premium Suite. If you want to implement this functionality, you should copy and paste code below into your app using code editor. Then compile and run your application. These C++ sample examples can be used in one or many applications.

ByteScout Premium Suite free trial version is available for download from our website. Free trial also includes programming tutorials along with 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](http://www.Bytescout.com)

Source Code Files:

```

#include "stdafx.h"
#include "comip.h"

#import "c:\\Program Files\\Bytescout PDF Extractor SDK\\net4.00\\Bytescout.PDFExtractor.dll"
using namespace Bytescout_PDFExtractor;

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

    // Create the interface pointer.
    _TextExtractorPtr pITextExtractor(__uuidof(TextExtractor));

    // Set the registration name and key
    // Note: You should use _bstr_t or BSTR to pass string to the library
    _bstr_t bstrRegName(L"DEMO");
    pITextExtractor->put_RegistrationName(bstrRegName);

    _bstr_t bstrRegKey(L"DEMO");
    pITextExtractor->put_RegistrationKey(bstrRegKey);

    // Load sample PDF document
    _bstr_t bstrPath(L"..\\..\\sample_ocr.pdf");
    pITextExtractor->LoadDocumentFromFile(bstrPath);

    // Enable Optical Character Recognition (OCR)
    // in .Auto mode (SDK automatically checks if needs to use OCR or not)
    pITextExtractor->put_OCMode(OCMode_Auto);

    // Set the location of OCR language data files
    _bstr_t bstrOCRLangDataPath(L"c:\\Program Files\\Bytescout PDF Extractor SDK\\net4.00\\Bytescout.PDFExtractor.Languages");
    pITextExtractor->put_OCRLanguageDataFolder(bstrOCRLangDataPath);

    // Set OCR language
    _bstr_t bstrOCRLanguage(L"eng");
    pITextExtractor->put_OCRLanguage(bstrOCRLanguage);

    // Set PDF document rendering resolution
    pITextExtractor->put_OCRResolution(300);

    // You can also apply various preprocessing filters
    // to improve the recognition on low-quality scans.

    _ImagePreprocessingFiltersCollection* pIImagePreprocessingFilters;
    pITextExtractor->get_OCRIImagePreprocessingFilters(&pIImagePreprocessingFilters);

    // Automatically deskew skewed scans
    //pIImagePreprocessingFilters->AddDeskew();

    // Remove vertical or horizontal lines (sometimes helps to avoid OCR errors)
    //pIImagePreprocessingFilters->AddVerticalLinesRemover();
    //pIImagePreprocessingFilters->AddHorizontalLinesRemover();
}

```

```

        // Repair broken letters
        //pIImagePreprocessingFilters->AddDilate();

        // Remove noise
        //pIImagePreprocessingFilters->AddMedian();

        // Apply Gamma Correction
        //pIImagePreprocessingFilters->AddGammaCorrection();

        // Add Contrast
        //pIImagePreprocessingFilters->AddContrast(20);

        // (!) You can use new OCRAnalyser class to find an optimal set of image
        // filters for your specific document.
        // See "OCR Analyser" example.

        // Save extracted text to file
        _bstr_t bstrOutputFile(L"output.txt");
        pITextExtractor->SaveTextToFile(bstrOutputFile);

        pITextExtractor->Release();

        CoUninitialize();

        return 0;
}

```

CPPEExample.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}") = "CPPEExample", "CPPEExample.vcxproj"
EndProject
Global
    GlobalSection(SolutionConfigurationPlatforms) = preSolution
        Debug|Win32 = Debug|Win32
        Release|Win32 = Release|Win32
    EndGlobalSection
    GlobalSection(ProjectConfigurationPlatforms) = postSolution
        {74A23FC0-E323-4980-9363-41326A457785}.Debug|Win32.ActiveCfg = Debug|Win32
        {74A23FC0-E323-4980-9363-41326A457785}.Debug|Win32.Build.0 = Debug|Win32
        {74A23FC0-E323-4980-9363-41326A457785}.Release|Win32.ActiveCfg = Release|Win32
        {74A23FC0-E323-4980-9363-41326A457785}.Release|Win32.Build.0 = Release|Win32
    EndGlobalSection

```

```
EndGlobalSection
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
EndGlobal
```

stdafx.cpp

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

