

How to extract PDF pages in C++ using ByteScout PDF Extractor SDK

The tutorial shows how to extract PDF pages in C++

Learn how to extract PDF pages in C++ with this source code sample. ByteScout PDF Extractor SDK: the SDK that helps developers to extract data from unstructured documents, pdf, images, scanned and electronic forms. Includes AI functions like automatic table detection, automatic table extraction and restructuring, text recognition and text restoration from pdf and scanned documents. Includes PDF to CSV, PDF to XML, PDF to JSON, PDF to searchable PDF functions as well as methods for low level data extraction. It can extract PDF pages in C++.

This rich sample source code in C++ for ByteScout PDF Extractor SDK includes the number of functions and options you should do calling the API to extract PDF pages. Follow the instructions from the scratch to work and copy the C++ code. Code testing will allow the function to be tested and work properly with your data.

You can download free trial version of ByteScout PDF Extractor SDK 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 PDF Extractor SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout PDF Extractor SDK](#)

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

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

Source Code Files:

```

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

// you may also refer to the tlb from \net4.00\ folder
// you may also want to include the tlb file into the project so you could compile
#import "c:\Program Files\Bytescout PDF Extractor SDK\net2.00\Bytescout.PDFExtractor.tlb"

using namespace Bytescout_PDFExtractor;

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

    // Create the interface pointer.
    _DocumentSplitterPtr pIDocumentSplitter(__uuidof(DocumentSplitter));

    // 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");
    pIDocumentSplitter->put_RegistrationName(bstrRegName);

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

    // you may enable optimization for extracted pages from documents
    // pIDocumentSplitter->put_OptimizeSplittedDocuments = true;

    // Load sample PDF document
    HRESULT sRes = S_OK;
    //1. extract selected pages (!note: page numbers are 1-based)
    _bstr_t bstrPath(L"..\\..\\sample2.pdf");
    _bstr_t bstrParam(L"page2.pdf");
    sRes = pIDocumentSplitter->ExtractPage(bstrPath, bstrParam, 2);

    // 2. split the doc into 2 parts at page #2
    // (!) Note: page numbers are 1-based
    _bstr_t bstrPathInput(L"..\\..\\sample2.pdf");
    _bstr_t bstrParam1(L"part1.pdf");
    _bstr_t bstrParam2(L"part2.pdf");
    sRes = pIDocumentSplitter->Split(bstrPathInput, bstrParam1, bstrParam2);

    // 3. merge page 2 extracted on step 1 along with base pdf
    // Create the interface pointer.
    _DocumentMergerPtr pIDocumentMerger(__uuidof(DocumentMerger));
    //_bstr_t bstrRegName(L"DEMO");
    pIDocumentMerger->put_RegistrationName(bstrRegName);
    //_bstr_t bstrRegKey(L"DEMO");
    pIDocumentMerger->put_RegistrationKey(bstrRegKey);

    // merge 2 files into the 3rd one
    _bstr_t bstrParamMerge1(L"page2.pdf");
    _bstr_t bstrParamMerge2(L"..\\..\\sample2.pdf");
    _bstr_t bstrParamMergeOutput(L"merged.pdf");

```

```

        sRes = pIDocumentMerger->Merge2(bstrParamMerge1, bstrParamMerge2,bstrParamMerge3);

        // finally release both instances
        pIDocumentSplitter->Release();
        pIDocumentMerger->Release();

        // uninitialized ActiveX COM support
        CoUninitialize();

        return 0;
    }

```

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 WinSxS.h
// set the _WIN32_WINNT macro to the platform you wish to support before including SDKDDKVer.h

#include <SDKDDKVer.h>
```

VIDEO

https://www.youtube.com/watch?v=s28W3_KMraU

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout PDF Extractor SDK Home Page](#)
[Explore ByteScout PDF Extractor SDK Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout PDF Extractor SDK 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)

