

How to split PDF document with PDF extractor SDK in C++ and ByteScout Premium Suite

Continuous learning is a crucial part of computer science and this tutorial shows how to split PDF document with PDF extractor SDK in C++

An easy to understand sample source code to learn how to split PDF document with PDF extractor SDK in 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. It can split PDF document with PDF extractor SDK in C++.

The SDK samples given below describe how to quickly make your application do split PDF document with PDF extractor SDK in C++ with the help of ByteScout Premium Suite. Just copy and paste the code into your C++ application's code and follow the instructions. Use of ByteScout Premium Suite in C++ is also described in the documentation included along with the product.

You can download free trial version of ByteScout Premium Suite from our website with this and other 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 "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 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)

