

# PDF OCR (optical character recognition) in VBScript and ByteScout PDF Extractor SDK

## How to use ByteScout PDF Extractor SDK for PDF OCR (optical character recognition) in VBScript

Writing of the code to PDF OCR (optical character recognition) in VBScript can be done by developers of any level using ByteScout PDF Extractor SDK. PDF OCR (optical character recognition) in VBScript can be implemented with ByteScout PDF Extractor SDK. ByteScout PDF Extractor SDK is the SDK is designed to help developers with pdf tables and pdf data extraction from unstructured documents like pdf, tiff, scans, images, scanned and electronic forms. The library is powered by OCR, computer vision and AI to provide unique functionality like table detection, automatic table structure extraction, data restoration, data restructuring and reconstruction. Supports PDF, TIFF, PNG, JPG images as input and can output CSV, XML, JSON formatted data. Includes full set of utilities like pdf splitter, pdf merger, searchable pdf maker and other utilities.

The SDK samples like this one below explain how to quickly make your application do PDF OCR (optical character recognition) in VBScript with the help of ByteScout PDF Extractor SDK. VBScript sample code is all you need: copy and paste the code to your VBScript application's code editor, add a reference to ByteScout PDF Extractor SDK (if you haven't added yet) and you are ready to go! Enhanced documentation and tutorials are available along with installed ByteScout PDF Extractor SDK if you'd like to dive deeper into the topic and the details of the API.

Trial version can be obtained from our website for free. It includes this and other source code samples for VBScript.

VBScript - ExtractTextFromImageOrScannedPdfUsingOcr.vbs

```
' This example demonstrates the use of Optical Character Recognition (OCR) to extract
text
' from scanned PDF documents and raster images.

' Create TextExtractor object
Set extractor = CreateObject("Bytescout.PDFExtractor.TextExtractor")
extractor.RegistrationName = "demo"
extractor.RegistrationKey = "demo"

' Load sample PDF document
extractor.LoadDocumentFromFile("../..\sample_ocr.pdf")

' Enable Optical Character Recognition (OCR)
extractor.OCRMode = 1 ' OCRMode.Auto = 1
```

```
' Set the location of OCR language data files
extractor.OCRLanguageDataFolder = "c:\Program Files\Bytescout PDF Extractor
SDK\ocrdata"

' Set OCR language
' "eng" for english, "deu" for German, "fra" for French, "spa" for Spanish etc -
according to files in "ocrdata" folder.
extractor.OCRLanguage = "eng"
' Find more language files at https://github.com/bytescout/ocrdata

' Set PDF document rendering resolution
extractor.OCRResolution = 300

' You can also apply various preprocessing filters to improve the recognition on low-
quality scans.
' But they significantly hit the performance, so do not enable them by default.

' Automatically deskew skewed scans
'extractor.OCRImagePreprocessingFilters.AddDeskew()

' Remove vertical or horizontal lines (sometimes helps to avoid OCR engine's page
segmentation errors)
'extractor.OCRImagePreprocessingFilters.AddVerticalLinesRemover()
'extractor.OCRImagePreprocessingFilters.AddHorizontalLinesRemover()

' Repair broken letters
'extractor.OCRImagePreprocessingFilters.AddDilate()

' Remove noise
'extractor.OCRImagePreprocessingFilters.AddMedian()

' Apply Gamma Correction
'extractor.OCRImagePreprocessingFilters.AddGammaCorrection()

' Add Contrast
' extractor.OCRImagePreprocessingFilters.AddContrast(20)

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

' Save extracted text to file
extractor.SaveTextToFile("output.txt")

WScript.Echo "Extracted text saved as 'output.txt'."

Set extractor = Nothing
```

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout PDF Extractor SDK](#)

[Explore documentation](#)

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

or

[Get Your Free API Key for www.PDF.co Web API](#)