

OCR with best dataset in VB.NET using ByteScout PDF Extractor SDK

How to code OCR with best dataset in VB.NET: How-To tutorial

The sample source codes on this page will demonstrate you how to make OCR with best dataset in VB.NET. OCR with best dataset in VB.NET 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.

VB.NET code snippet like this for ByteScout PDF Extractor SDK works best when you need to quickly implement OCR with best dataset in your VB.NET application. To do OCR with best dataset in your VB.NET project or application you may simply copy & paste the code and then run your app! You can use these VB.NET sample examples in one or many applications.

ByteScout PDF Extractor SDK 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 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:

```
Imports Bytescout.PDFExtractor
```

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

```
' To make OCR work you should add the following references to your project:
' "Bytescout.PDFExtractor.dll", "Bytescout.PDFExtractor.OCRExtension.dll".
```

```
Class Program
```

```
    Friend Shared Sub Main(args As String())
```

```
        ' Create Bytescout.PDFExtractor.TextExtractor instance
```

```
        Dim extractor As New TextExtractor()
```

```
        extractor.RegistrationName = "demo"
```

```
        extractor.RegistrationKey = "demo"
```

```
        ' Load sample PDF document
```

```
        extractor.LoadDocumentFromFile("sample_ocr.pdf")
```

```
        ' Enable Optical Character Recognition (OCR)
```

```
        ' in .Auto mode (SDK automatically checks if needs to use OCR or not)
```

```
        extractor.OCRMode = OCRMode.Auto
```

```
        ' Set the location of OCR language data files
```

```
        extractor.OCRLanguageDataFolder = "c:\Program Files\Bytescout PDF Extractor SDK\Languages"
```

```
        ' Set OCR language
```

```
        extractor.OCRLanguage = "eng" ' "eng" for english, "deu" for German, "fra" for French
```

```
        ' Find more language files at https://github.com/bytescout/ocrdata/tree/master
```

```
        ' Set PDF document rendering resolution
```

```
        extractor.OCRResolution = 300
```

```
        ' You can also apply various preprocessing filters
```

```
        ' to improve the recognition on low-quality scans.
```

```
        ' Automatically deskew skewed scans
```

```
        extractor.OCRImagePreprocessingFilters.AddDeskew()
```

```
        ' Remove vertical or horizontal lines (sometimes helps to avoid OCR engine's problems)
```

```
        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 preproc
' filters for your specific document.
' See "OCR Analyser" example.

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

' Cleanup
extractor.Dispose()

' Open output file in default associated application
System.Diagnostics.Process.Start("output.txt")

End Sub

End Class
```

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)

