

## ocr with mean dataset with pdf extractor sdk in VB.NET and ByteScout PDF Suite

Learn to code in VB.NET to make ocr with mean dataset with pdf extractor sdk with this simple How-To tutorial

This page explains the steps and algorithm of implementing ocr with mean dataset with pdf extractor sdk and how to make it work in your application. ByteScout PDF Suite helps with ocr with mean dataset with pdf extractor sdk in VB.NET. ByteScout PDF Suite is the set that includes 6 SDK products to work with PDF from generating rich PDF reports to extracting data from PDF documents and converting them to HTML. This bundle includes PDF (Generator) SDK, PDF Renderer SDK, PDF Extractor SDK, PDF to HTML SDK, PDF Viewer SDK and PDF Generator SDK for Javascript.

VB.NET code snippet like this for ByteScout PDF Suite works best when you need to quickly implement ocr with mean dataset with pdf extractor sdk in your VB.NET application. If you want to know how it works, then this VB.NET sample code should be copied and pasted into your application's code editor. Then just compile and run it. Enjoy writing a code with ready-to-use sample VB.NET codes to implement ocr with mean dataset with pdf extractor sdk using ByteScout PDF Suite.

On our website you may get trial version of ByteScout PDF Suite for free. Source code samples are included to help you with your VB.NET application.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout PDF Suite](#)

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

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

Source Code Files:

## Program.vb

```
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"
```

```
        ' 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 prepro
' 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=NEwNs2b9YN8>

## ON-PREMISE OFFLINE SDK

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

[www.bytescout.com](http://www.bytescout.com)