

ocr with fast dataset with pdf extractor sdk in VB.NET using ByteScout PDF Suite

Build ocr with fast dataset with pdf extractor sdk in VB.NET

:

Step-by-step instructions on how to do ocr with fast dataset with pdf extractor sdk in VB.NET

The samples of source code documentation give a quick and simple method to apply a required functionality into your application. ByteScout PDF Suite was created to assist ocr with fast 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.

This rich and prolific sample source code in VB.NET for ByteScout PDF Suite contains various functions and options you should do calling the API to implement ocr with fast dataset with pdf extractor sdk. To use ocr with fast dataset with pdf extractor sdk in your VB.NET project or application just copy & paste the code and then run your app! VB.NET application implementation mostly involves various stages of the software development so even if the functionality works please check it with your data and the production environment.

ByteScout PDF Suite is available as a free trial. You may get it from our website along with all other source code samples for VB.NET applications.

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.OCRIImagePreprocessingFilters.AddMedian()

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

' Add Contrast
    ' extractor.OCRIImagePreprocessingFilters.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