ocr (optical character recognition) and pdf with pdf extractor sdk in VB.NET using ByteScout PDF Suite

Build ocr (optical character recognition) and pdf with pdf extractor sdk in VB.NET

:

Step-by-step instructions on how to do ocr (optical character recognition) and pdf with pdf extractor sdk in VB.NET

An easy to understand guide to learn how to ocr (optical character recognition) and pdf with pdf extractor sdk in VB.NET. ByteScout PDF Suite was created to assist ocr (optical character recognition) and pdf with pdf extractor sdk in VB.NET. ByteScout PDF Suite is the bundle that provides six different SDK libraries 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 (optical character recognition) and pdf with pdf extractor sdk. Follow the steps-by-step instructions from the scratch to work and copy and paste code for VB.NET into your editor. This basic programming language sample code for VB.NET will do the whole work for you in implementing ocr (optical character recognition) and pdf with pdf extractor sdk in your app.

If you want to try other samples for VB.NET then free trial version of ByteScout PDF Suite is available on our website.

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

Source Code Files:

Program.vb

```
Imports Bytescout.PDFExtractor

' This example demonstrates the use of Optical Character Recognition (OCR) to extract t
' 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 SD

        ' Set OCR language
        extractor.OCRLanguage = "eng"  ' "eng" for english, "deu" for German, "fra" fo
        ' 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.

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

        ' Remove vertical or horizontal lines (sometimes helps to avoid OCR engine's p
        '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=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