repair text in PDF in C# using ByteScout PDF Extractor SDK

How to use ByteScout PDF Extractor SDK for repair text in PDF in C#

Sample source codes below will show you how to cope with a difficult task, for example, repair text in PDF in C#. ByteScout PDF Extractor SDK was made to help with repair text in PDF in C#. ByteScout PDF Extractor SDK is the Software Development Kit (SDK) that is designed to help developers with 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.

Fast application programming interfaces of ByteScout PDF Extractor SDK for C# plus the instruction and the C# code below will help you quickly learn repair text in PDF. To do repair text in PDF in your C# project or application you may simply copy & paste the code and then run your app! Further enhancement of the code will make it more vigorous.

Our website provides free trial version of ByteScout PDF Extractor SDK. It comes along with all these source code samples with the goal to help you with your C# application implementation.

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

Source Code Files:

Program.cs

```
using Bytescout.PDFExtractor;
using System;

namespace ReadTextFromNoisyImage
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                //Read all text from noisy image file
                using (TextExtractor extractor = new TextExtractor())
                {
                    // Load noisy image document
                    extractor.LoadDocumentFromFile("sample.png");

                    // Set the font repairing OCR mode
                    extractor.OCRMode = OCRMode.TextFromImagesAndVectorsAndRepairedFon

                    // Set the location of OCR language data files
                    extractor.OCRLanguageDataFolder = @"c:\Program Files\Bytescout PDF

                    // Set OCR language
                    extractor.OCRLanguage = "eng"; // "eng" for english, "deu" for Ger
                                                   // Find more language files at http

                    // Set document rendering resolution
                    extractor.OCRResolution = 300;

                    // You can also apply various preprocessing filters
                    // to improve the recognition on low-quality scans.

                                        Console.WriteLine("Please wait while PDF Extrac

                    // Automatically deskew skewed scans
                    extractor.OCRImagePreprocessingFilters.AddDeskew();

                    // Remove vertical or horizontal lines (sometimes helps to avoid O
                    //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 OCRAnalyser class to find an optimal set of
                    // filters for your specific document.
                    // See "OCR Analyser" example.

                    //Read all text
                    string allText = extractor.GetText();

                    Console.Clear();
                    Console.WriteLine("Extracted Text: \n\n" + allText);
                }
            }
            catch (Exception ex)
            {
                Console.Clear();
                Console.WriteLine("Exception: " + ex.Message);
            }

            Console.ReadLine();
        }
    }
}
```

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]()

[visit www.PDF.co]()

www.bytescout.com