www.bytescout.com

How to convert scanned PDF to XML with PDF extractor SDK in C# and ByteScout Data Extraction Suite

Learning is essential in computer world and the tutorial below will demonstrate how to convert scanned PDF to XML with PDF extractor SDK in C#

ByteScout simple and easy to understand tutorials are planned to describe the code for both C# beginners and advanced programmers. Want to convert scanned PDF to XML with PDF extractor SDK in your C# app? ByteScout Data Extraction Suite is designed for it. ByteScout Data Extraction Suite is the set that includes 3 SDK products for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK.

The following code snippet for ByteScout Data Extraction Suite works best when you need to quickly convert scanned PDF to XML with PDF extractor SDK in your C# application. Simply copy and paste in your C# project or application you and then run your app! Applying C# application mostly includes various stages of the software development so even if the functionality works please test it with your data and the production environment.

ByteScout Data Extraction Suite free trial version is available on our website. C# and other programming languages are supported.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout Data Extraction Suite

Explore API Documentation

Get Free Training for ByteScout Data Extraction Suite

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Program.cs

```
using System.Diagnostics;
using Bytescout.PDFExtractor;

// This example demonstrates the use of Optical Character Recognition (OCR) to extract
// 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'.

namespace ScannedPdfToXML
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create Bytescout.PDFExtractor.XMLExtractor instance
            XMLExtractor extractor = new XMLExtractor();
            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 Extracto

            // Set OCR language
            extractor.OCRLanguage = "eng"; // "eng" for english, "deu" for German, "fr
            // 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
            //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 image pr
            // filters for your specific document.
            // See "OCR Analyser" example.


            // Save extracted text to file
            extractor.SaveXMLToFile("output.xml");

            // Cleanup
            extractor.Dispose();

            // Open result document in default associated application (for demo purpose
            ProcessStartInfo processStartInfo = new ProcessStartInfo("output.xml");
            processStartInfo.UseShellExecute = true;
            Process.Start(processStartInfo);
        }
    }
}
```

VIDEO

https://www.youtube.com/watch?v=NEwNs2b9YN8

ON-PREMISE OFFLINE SDK

60 Day Free Trial or Visit ByteScout Data Extraction Suite Home Page
Explore ByteScout Data Extraction Suite Documentation
Explore Samples
Sign Up for ByteScout Data Extraction Suite 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