How to reduce memory usage for PDF to text in C# with ByteScout PDF Extractor SDK

This code in C# shows how to reduce memory usage for PDF to text with this how to tutorial

We made thousands of pre-made source code pieces for easy implementation in your own programming projects. What is ByteScout PDF Extractor SDK? It 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. It can help you to reduce memory usage for PDF to text in your C# application.

You will save a lot of time on writing and testing code as you may just take the C# code from ByteScout PDF Extractor SDK for reduce memory usage for PDF to text below and use it in your application. In order to implement the functionality, you should copy and paste this code for C# below into your code editor with your app, compile and run your application. This basic programming language sample code for C# will do the whole work for you to reduce memory usage for PDF to text.

Our website provides trial version of ByteScout PDF Extractor SDK for free. It also includes documentation and source code samples.

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 System;
using System.IO;
using System.Diagnostics;
using Bytescout.PDFExtractor;

namespace ReduceMemoryUsage
{
    class Program
    {
        static void Main(string[] args)
        {
            // When processing huge PDF documents you may run into OutOfMemoryExceptio
            // This example demonstrates a way to spare the memory by disabling page d

            // Create Bytescout.PDFExtractor.TextExtractor instance
            using (TextExtractor extractor = new TextExtractor("demo", "demo"))
            {
                try
                {
                    // Load sample PDF document
                    extractor.LoadDocumentFromFile("sample2.pdf");

                    // Disable page data caching, so processed pages wiil be disposed
                    extractor.PageDataCaching = PageDataCaching.None;

                    // Save extracted text to file
                    extractor.SaveTextToFile("output.txt");
                }
                catch (PDFExtractorException exception)
                {
                    Console.Write(exception.ToString());
                }
            }

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

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