www.bytescout.com

How to reduce memory usage for PDF to text with PDF extractor SDK in C# using ByteScout Data Extraction Suite

Learn to reduce memory usage for PDF to text with PDF extractor SDK in C#

Sample source code below will display you how to manage a complex task like reduce memory usage for PDF to text with PDF extractor SDK in C#. What is ByteScout Data Extraction Suite? It 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. It can help you to reduce memory usage for PDF to text with PDF extractor SDK in your C# application.

The following code snippet for ByteScout Data Extraction Suite works best when you need to quickly reduce memory usage for PDF to text with PDF extractor SDK in your C# application. This C# sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! This basic programming language sample code for C# will do the whole work for you to reduce memory usage for PDF to text with PDF extractor SDK.

Trial version of ByteScout Data Extraction Suite is available for free. Source code samples are included to help you with your C# app.

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 purpos
            ProcessStartInfo processStartInfo = new ProcessStartInfo("output.txt");
            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