How to reduce memory usage for pdf extraction with pdf extractor sdk in VBScript with ByteScout Data Extraction Suite

This code in VBScript shows how to reduce memory usage for pdf extraction with pdf extractor sdk with this how to tutorial

This sample source code below will display you how to reduce memory usage for pdf extraction with pdf extractor sdk in VBScript. Want to reduce memory usage for pdf extraction with pdf extractor sdk in your VBScript 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 reduce memory usage for pdf extraction with pdf extractor sdk in your VBScript application. Just copy and paste the code into your VBScript application's code and follow the instructions. Complete and detailed tutorials and documentation are available along with installed ByteScout Data Extraction Suite if you'd like to learn more about the topic and the details of the API.

ByteScout Data Extraction Suite free trial version is available on our website. VBScript 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:

ReduceMemoryUsage.vbs

```vbs
' When processing huge PDF documents you may run into OutOfMemoryException.
' This example demonstrates a way to spare the memory by disabling page data caching.

' Create Bytescout.PDFExtractor.TextExtractor object
Set extractor = CreateObject("Bytescout.PDFExtractor.TextExtractor")
extractor.RegistrationName = "demo"
extractor.RegistrationKey = "demo"

' Load sample PDF document
extractor.LoadDocumentFromFile("..\..\sample2.pdf")

' Disable page data caching, so processed pages wiil be disposed automatically
extractor.PageDataCaching = 0 ' 0 - no caching; 1 - cache all pages.

' Save extracted text to file
extractor.SaveTextToFile("output.txt")

Set extractor = Nothing

WScript.Echo "Extracted data saved to 'output.txt' file."
```

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