How to reduce memory usage for PDF extraction in VBScript and ByteScout PDF Extractor SDK

This tutorial will show how to reduce memory usage for PDF extraction in VBScript

Reduce memory usage for PDF extraction is easy to implement in VBScript if you use these source codes below. ByteScout PDF Extractor SDK is the SDK that helps developers to extract data from unstructured documents, pdf, images, scanned and electronic forms. Includes AI functions like automatic table detection, automatic table extraction and restructuring, text recognition and text restoration from pdf and scanned documents. Includes PDF to CSV, PDF to XML, PDF to JSON, PDF to searchable PDF functions as well as methods for low level data extraction. It can be used to reduce memory usage for PDF extraction using VBScript.

This code snippet below for ByteScout PDF Extractor SDK works best when you need to quickly reduce memory usage for PDF extraction in your VBScript application. In order to implement the functionality, you should copy and paste this code for VBScript below into your code editor with your app, compile and run your application. Use of ByteScout PDF Extractor SDK in VBScript is also explained in the documentation included along with the product.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with 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:

ReduceMemoryUsage.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=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