How to reduce memory usage during PDF to HTML conversion in VB.NET with ByteScout PDF To HTML SDK

How to reduce memory usage during PDF to HTML conversion in VB.NET

Sample source code below will show you how to cope with a difficult task like reduce memory usage during PDF to HTML conversion in VB.NET. ByteScout PDF To HTML SDK is the SDK for PDF to HTML conversin. Generates static HTML version of input PDF and reconstructs all visual layout, text positions, vector and raster drawings, images, and form elements. Generated HTML requires no special software and can be viewed in any Internet browser. It can reduce memory usage during PDF to HTML conversion in VB.NET.

You will save a lot of time on writing and testing code as you may just take the VB.NET code from ByteScout PDF To HTML SDK for reduce memory usage during PDF to HTML conversion below and use it in your application. In your VB.NET project or application you may simply copy & paste the code and then run your app! Enjoy writing a code with ready-to-use sample VB.NET codes.

You can download free trial version of ByteScout PDF To HTML SDK from our website to see and try many others source code samples for VB.NET.

FOR MORE INFORMATION AND FREE TRIAL:

Download Free Trial SDK (on-premise version)

Read more about ByteScout PDF To HTML SDK

Explore API Documentation

Get Free Training for ByteScout PDF To HTML SDK

Get Free API key for Web API

visit www.ByteScout.com

Source Code Files:

Program.vb

```vb
Imports Bytescout.PDF2HTML

Class Program
    Friend Shared Sub Main(args As String())

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

        ' Create Bytescout.PDF2HTML.HTMLExtractor instance
        Using extractor As New HTMLExtractor("demo", "demo")

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

                ' Disable page data caching, so processed pages will be disposed autom
                extractor.PageDataCaching = PageDataCaching.None

                ' Save result to file
                extractor.SaveHtmlToFile("output.html")

            Catch exception As PDF2HTMLException
                Console.Write(exception.ToString())
            End Try

        End Using

        ' Open the output file in default associated application
        System.Diagnostics.Process.Start("output.txt")

    End Sub
End Class
```

VIDEO

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout PDF To HTML SDK Home Page](#)
[Explore ByteScout PDF To HTML SDK Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout PDF To HTML 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](#)