

How to reduce memory usage with pdf extractor sdk in VB.NET using ByteScout Premium Suite

If you want to learn more then this tutorial will show how to reduce memory usage with pdf extractor sdk in VB.NET

These source code samples are assembled by their programming language and functions they apply. What is ByteScout Premium Suite? It is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording. It can help you to reduce memory usage with pdf extractor sdk in your VB.NET application.

The SDK samples given below describe how to quickly make your application do reduce memory usage with pdf extractor sdk in VB.NET with the help of ByteScout Premium Suite. Simply copy and paste in your VB.NET project or application you and then run your app! Applying VB.NET application mostly includes various stages of the software development so even if the functionality works please test it with your data and the production environment.

ByteScout Premium Suite free trial version is available on our website. VB.NET and other programming languages are supported.

FOR MORE INFORMATION AND FREE TRIAL:

[Download Free Trial SDK \(on-premise version\)](#)

[Read more about ByteScout Premium Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Premium Suite](#)

[Get Free API key for Web API](#)

[visit www.Bytescout.com](http://www.Bytescout.com)

Source Code Files:

```
Imports Bytescout.PDFExtractor
Imports System.IO

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 caching.

        ' Create Bytescout.PDFExtractor.TextExtractor instance
        Using extractor As New TextExtractor("demo", "demo")

            Try

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

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

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

                Catch exception As PDFExtractorException
                    Console.WriteLine(exception.ToString())
                End Try

            End Using

            ' Open result file in default associated application (for demo purposes)
            System.Diagnostics.Process.Start("output.txt")

        End Sub

    End Class
```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

[60 Day Free Trial](#) or [Visit ByteScout Premium Suite Home Page](#)
[Explore ByteScout Premium Suite Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Premium Suite Online Training](#)

ON-DEMAND REST WEB API

[Get Your API Key](#)
[Explore Web API Docs](#)
[Explore Web API Samples](#)

[visit www.ByteScout.com](http://www.ByteScout.com)

[visit www.PDF.co](http://www.PDF.co)

www.bytescout.com