

## How to reduce memory usage with pdf extractor sdk in VB.NET and ByteScout PDF Suite

Learn to code in VB.NET to reduce memory usage with pdf extractor sdk with this step-by-step tutorial

This sample source code below will display you how to reduce memory usage with pdf extractor sdk in VB.NET. ByteScout PDF Suite: the set that includes 6 SDK products to work with PDF from generating rich PDF reports to extracting data from PDF documents and converting them to HTML. This bundle includes PDF (Generator) SDK, PDF Renderer SDK, PDF Extractor SDK, PDF to HTML SDK, PDF Viewer SDK and PDF Generator SDK for Javascript. It can reduce memory usage with pdf extractor sdk in VB.NET.

Want to quickly learn? This fast application programming interfaces of ByteScout PDF Suite for VB.NET plus the guidelines and the code below will help you quickly learn how to reduce memory usage with pdf extractor sdk. IF you want to implement the functionality, just copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. Want to see how it works with your data then code testing will allow the function to be tested and work properly.

Our website gives trial version of ByteScout PDF Suite for free. It also includes documentation and source code samples.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Suite](#)

[Explore API Documentation](#)

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

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

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

Source Code Files:

## Program.vb

```
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 PDF Suite Home Page](#)  
[Explore ByteScout PDF Suite Documentation](#)  
[Explore Samples](#)  
[Sign Up for ByteScout PDF 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](http://www.bytescout.com)