

How to reduce memory usage in VB.NET with ByteScout PDF Extractor SDK

This tutorial will show how to reduce memory usage in VB.NET

ByteScout tutorials are designed to explain the code for both VB.NET beginners and advanced programmers. 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 using 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 Extractor SDK for reduce memory usage below and use it in your application. In order to implement the functionality, you should copy and paste this code for VB.NET below into your code editor with your app, compile and run your application. Test VB.NET sample code examples whether they respond your needs and requirements for the project.

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

VB.NET - 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

            End Try

        End Using

    End Sub

End Class
```

```
' 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
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Extractor SDK](#)

[Explore documentation](#)

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

or

[Get Your Free API Key for www.PDF.co Web API](#)