

## How to find keyword in pdf and extract page with pdf extractor sdk in VB.NET using ByteScout PDF Suite

This code in VB.NET shows how to find keyword in pdf and extract page with pdf extractor sdk with this how to tutorial

This sample source code below will display you how to find keyword in pdf and extract page with pdf extractor sdk in VB.NET. Want to find keyword in pdf and extract page with pdf extractor sdk in your VB.NET app? ByteScout PDF Suite is designed for it. ByteScout PDF Suite is 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.

These VB.NET code samples for VB.NET guide developers to speed up coding of the application when using ByteScout PDF Suite. 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. Check VB.NET sample code samples to see if they respond to your needs and requirements for the project.

The trial version of ByteScout PDF Suite can be downloaded for free from our website. It also includes source code samples for VB.NET and other programming languages.

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

```
' This example page extraction by found keyword.

Imports Bytescout.PDFExtractor

Class Program

    Friend Shared Sub Main(args As String())

        Dim inputFile As String = ".\sample2.pdf"

        ' Create Bytescout.PDFExtractor.TextExtractor instance
        Dim extractor As New TextExtractor()
        extractor.RegistrationName = "demo"
        extractor.RegistrationKey = "demo"

        ' Load sample PDF document
        extractor.LoadDocumentFromFile(inputFile)

        Dim pageCount As Integer = extractor.GetPageCount()

        ' Search each page for a keyword
        For i As Integer = 0 To pageCount - 1

            If extractor.Find(i, "bombardment", False) Then

                ' Extract page
                Using splitter As New DocumentSplitter("demo", "demo")

                    splitter.OptimizeSplittedDocuments = True

                    Dim pageNumber As Integer = i + 1
                    ' (!) page number in ExtractPage() is 1-based
                    Dim outputFile As String = ".\page" & pageNumber.ToString() & ".pdf"
                    splitter.ExtractPage(inputFile, outputFile, pageNumber)

                    Console.WriteLine("Extracted page " & pageNumber.ToString() & " to " & outputFile)

                End Using
            End If
        Next

        ' Cleanup
        extractor.Dispose()

        Console.WriteLine()
        Console.WriteLine("Press any key...")
        Console.ReadKey()

    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)