

How to split PDF document by found barcode in VB.NET with ByteScout BarCode Reader SDK

The tutorial shows how to split PDF document by found barcode in VB.NET

This sample source code below will demonstrate you how to split PDF document by found barcode in VB.NET. What is ByteScout BarCode Reader SDK? It is the SDK for reading of barcodes from PDF, images and live camera or video. Almost every common type like Code 39, Code 128, GS1, UPC, QR Code, Datamatrix, PDF417 and many others are supported. Supports noisy and defective images and docs. Includes optional documents splitter and merger for pdf and tiff based on found barcodes. Batch mode is supported for superior performance using multiple threads. Decoded values are easily exported to JSON, CSV, XML and to custom format. It can help you to split PDF document by found barcode in your VB.NET application.

This code snippet below for ByteScout BarCode Reader SDK works best when you need to quickly split PDF document by found barcode in your VB.NET application. This VB.NET sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Code testing will allow the function to be tested and work properly with your data.

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

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout BarCode Reader SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout BarCode Reader SDK](#)

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

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

Source Code Files:

Module1.vb

```
Imports System.IO
Imports System.Text
Imports Bytescout.BarCodeReader

Module Module1

    Sub Main()

        Const inputFile As String = "Barcodes.pdf"

        Console.WriteLine("Reading barcode(s) from image {0}", Path.GetFullPath(inputFile))

        ' Create Bytescout.BarCodeReader.Reader instance
        Dim reader As New Reader()
        reader.RegistrationName = "demo"
        reader.RegistrationKey = "demo"

        ' Set barcode type to find
        reader.BarcodeTypesToFind.Code39 = True

        ' -----
        ' NOTE: We can read barcodes from specific page to increase performance .
        ' For sample please refer to "Decoding barcodes from PDF by pages" program.
        ' -----

        ' Find barcode in PDF document
        reader.ReadFrom(inputFile)

        ' Method 1: Split PDF document in two parts by found barcode
        ' NOTE: In Full version of the SDK this method is unlocked in "PRO" license type
        reader.SplitDocument("barcodes.pdf", "part1.pdf", "part2.pdf", reader.FoundBarcode)

        ' Method 2: Extract page containing the barcode from PDF document
        ' NOTE: In Full version of the SDK this method is unlocked in "PRO" license type
        reader.ExtractPageFromDocument("barcodes.pdf", "extracted_page.pdf", reader.FoundBarcode)

        ' Method 3: Split PDF document into parts in one pass.
        ' NOTE: In Full version of the SDK this method is unlocked in "PRO" license type

        Dim pageRanges As StringBuilder = New StringBuilder()

        ' Create string containing page ranges to extract in the form "1-4,6-8,10-11,12"
        For i As Integer = 0 To reader.FoundBarcodes.Length - 1

            Dim barcode As FoundBarcode = reader.FoundBarcodes(i)

            ' Add pages before the first barcode found
```

```

    If i = 0 And barcode.Page > 0 Then
        pageRanges.Append("1")
        If barcode.Page > 1 Then
            pageRanges.Append("-")
            pageRanges.Append(barcode.Page)
        End If
        pageRanges.Append(",")
    End If

    ' Add page with barcode
    pageRanges.Append(barcode.Page + 1) ' +1 because we skip the page with barcode

    ' Add range untill the next barcode
    If i < reader.FoundBarcodes.Length - 1 Then
        If reader.FoundBarcodes(i + 1).Page - barcode.Page > 1 Then
            pageRanges.Append("-")
            pageRanges.Append(reader.FoundBarcodes(i + 1).Page)
        End If
        pageRanges.Append(",")
    Else
        ' for the last found barcode add ending "-" meaning "to the last page"
        pageRanges.Append("-")
    End If
Next

' Split document
Dim splittedParts As String() = reader.SplitDocument("barcodes.pdf", pageRanges)
' The method returns array of file names. Rename files as desired.

' Cleanup
reader.Dispose()

For Each fileName As String In splittedParts
    Console.WriteLine(fileName)
Next

Console.WriteLine("Press enter key to exit...")
Console.ReadLine()

End Sub

End Module

```

VIDEO

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

ON-PREMISE OFFLINE SDK

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