

How to reduce CPU usage in VB.NET using ByteScout BarCode Reader SDK

The tutorial below will demonstrate how to reduce CPU usage in VB.NET

These source code samples are listed and grouped by their programming language and functions they use. ByteScout BarCode Reader SDK 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 barcodess. 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 be used to reduce CPU usage using VB.NET.

This code snippet below for ByteScout BarCode Reader SDK works best when you need to quickly reduce CPU usage in your VB.NET 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. Code testing will allow the function to be tested and work properly with your data.

You can download free trial version of ByteScout BarCode Reader SDK from our website to see and try many others source code samples for VB.NET.

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:

Program.vb

```
Imports Bytescout.BarCodeReader

Class Program
    Friend Shared Sub Main(args As String())

        ' Barcode reader instance
        Dim reader As New Reader()
        reader.RegistrationName = "demo"
        reader.RegistrationKey = "demo"

        'If you are reading barcodes from PDF Then you may reduce CPU And RAM load Using
        '- instead of using All1D Or All2D barcode types, set it to the specific types
        '- reduce PDF rendering resolution to 200-150 dpi (depends on your document)
        '- set specific pages to read barcodes from. If you have barcodes on 2 first pages
        '- if barcodes are always located / printed in the same corner then also specify area

        ' Input filename
        Dim inputFileName As String = "barcode_multipage.pdf"

        ' Set specific barcode type to read
        reader.BarcodeTypesToFind.Code128 = True

        ' Reduce PDF rendering resolution
        reader.PDFRenderingResolution = 150

        ' Set specific area to read from
        reader.CustomAreaLeft = 407
        reader.CustomAreaTop = 494
        reader.CustomAreaHeight = 605
        reader.CustomAreaWidth = 999

        ' Set specific page to read from along with filename
        reader.ReadFromPdfFilePage(inputFileName, 1, 1)

        ' Get all found barcodes
        Dim barcodes As FoundBarcode() = reader.FoundBarcodes

        ' Display found barcodes
        Console.WriteLine("Reading barcode(s) from PDF file...")

        For Each barcode As FoundBarcode In barcodes
            Console.WriteLine("Found Barcode - Type: '{0}', Value: '{1}'", barcode.Type, barcode.Value)
        Next

        ' Cleanup
        reader.Dispose()

        Console.WriteLine()
        Console.WriteLine("Press any key to continue.")
        Console.ReadKey()
    End Sub
End Class
```

VIDEO

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

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