

# parallel barcode decoding in VB.NET and ByteScout BarCode Reader SDK

## Tutorial: how to do parallel barcode decoding in VB.NET

Sample source codes below will show you how to cope with a difficult task, for example, parallel barcode decoding in VB.NET. ByteScout BarCode Reader SDK was made to help with parallel barcode decoding in VB.NET. 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.

VB.NET, code samples for VB.NET, developers help to speed up the application development and writing a code when using ByteScout BarCode Reader SDK. In order to implement this functionality, you should copy and paste code below into your app using code editor. Then compile and run your application. Use of ByteScout BarCode Reader SDK in VB.NET is also explained in the documentation included along with the product.

ByteScout BarCode Reader SDK free trial version is available for download from our website. Free trial also includes programming tutorials along with source code samples.

VB.NET - Module1.vb

```
Imports Bytescout.BarCodeReader
Imports System.Threading

Module Module1

    Const InputFile As String = ".\example.pdf"

    ' Limit to 4 threads in queue.
    ' Set this value to number of your processor cores for max performance.
    Dim ThreadLimiter As Semaphore = New Semaphore(4, 4)

    Sub Main()

        Const numberOfRuns As Integer = 10
        Dim doneEvents(numberOfRuns - 1) As ManualResetEvent

        For i As Integer = 0 To numberOfRuns - 1

            ' Wait for the queue
```

```

        ThreadLimiter.WaitOne()

        doneEvents(i) = New ManualResetEvent(False)

        ThreadPool.QueueUserWorkItem(New WaitCallback(AddressOf ThreadProc), New
Object() {i, doneEvents(i)})

    Next

    WaitHandle.WaitAll(doneEvents)

    Console.WriteLine("All threads done.")
    Console.WriteLine()

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

End Sub

Sub ThreadProc(ByVal stateInfo As Object)

    Dim threadIndex As Integer = stateInfo(0)
    Dim waitEvent As ManualResetEvent = stateInfo(1)

    Console.WriteLine("Thread #" & threadIndex & " started...")

    Try

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

        ' Set barcode type to find
        reader.BarcodeTypesToFind.Code128 = True

        ' Read barcodes
        Dim barcodes As FoundBarcode() = reader.ReadFrom(InputFile)

        ' Cleanup
        reader.Dispose()

        Console.WriteLine("Thread #" & threadIndex & " finished with " &
barcodes.Length & " barcodes found.")

    Catch ex As Exception

        Console.WriteLine("Thread #" & threadIndex & " failed with
exception:\r\n" & ex.Message)

    Finally

        ' Signal the thread is finished
        waitEvent.Set()

        ' Release semaphore
        ThreadLimiter.Release()

    End Try

End Sub

```

End Module

---

FOR MORE INFORMATION AND FREE TRIAL:

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

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

[Explore documentation](#)

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

or

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