

parallel processing of pdf files with pdf extractor sdk in VB.NET using ByteScout PDF Suite

How to apply ByteScout PDF Suite for parallel processing of pdf files with pdf extractor sdk in VB.NET

The example source codes on this page will display you how to make parallel processing of pdf files with pdf extractor sdk in VB.NET. Parallel processing of pdf files with pdf extractor sdk in VB.NET can be applied with ByteScout PDF Suite. ByteScout PDF Suite is the bundle that provides six different SDK libraries 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.

The below SDK samples describe how to quickly make your application do parallel processing of pdf files with pdf extractor sdk in VB.NET with the help of ByteScout PDF Suite. To use parallel processing of pdf files with pdf extractor sdk in your VB.NET project or application just copy & paste the code and then run your app! Enjoy writing a code with ready-to-use sample VB.NET codes to implement parallel processing of pdf files with pdf extractor sdk using ByteScout PDF Suite.

On our website you may get trial version of ByteScout PDF Suite for free. Source code samples are included to help you with your VB.NET application.

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:

Module1.vb

```
Imports System.IO
Imports System.Threading
Imports Bytescout.PDFExtractor

Module Module1

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

    <MTAThread(>
    Sub Main()

        ' Get all PDF files in a folder
        Dim files = Directory.GetFiles("..\\..\\..\\..\\", "*.pdf")
        ' Array of events to wait
        Dim doneEvents(files.Length - 1) As ManualResetEvent

        For i As Integer = 0 To files.Length - 1

            ' Wait for the queue
            ThreadLimiter.WaitOne()

            ' Start thread with filename and event in params
            doneEvents(i) = New ManualResetEvent(False)
            Dim threadData As Object = New Object() {files(i), doneEvents(i)}
            ThreadPool.QueueUserWorkItem(AddressOf ConvertPdfToTxt, threadData)

        Next

        ' Wait until all threads finish
        WaitHandle.WaitAll(doneEvents)

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

    End Sub

    Private Sub ConvertPdfToTxt(ByVal state As Object)

        ' Get filename and event from params
        Dim file As String = state(0)
        Dim doneEvent As ManualResetEvent = state(1)

        Dim resultFileName = Path.GetFileName(file) + ".txt"

        Try

            Console.WriteLine("Converting " + file)
```

```
Using extractor As New TextExtractor("demo", "demo")

    extractor.LoadDocumentFromFile(file)
    extractor.SaveTextToFile(resultFileName)

End Using

Console.WriteLine("Finished " + resultFileName)

Finally

    ' Signal the thread is finished
    doneEvent.Set()
    ' Release semaphore
    ThreadLimiter.Release()

End Try

End Sub

End Module
```

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