

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

parallel processing of pdf files with pdf extractor sdk in VB.NET

The documentation is crafted to assist you to apply the features on your side easily. ByteScout Data Extraction Suite was created to assist parallel processing of pdf files with pdf extractor sdk in VB.NET. ByteScout Data Extraction Suite is the set that includes 3 SDK products for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK.

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 Data Extraction Suite. Follow the steps-by-step instructions from the scratch to work and copy and paste code for VB.NET into your editor. Enjoy writing a code with ready-to-use sample VB.NET codes to add parallel processing of pdf files with pdf extractor sdk functions using ByteScout Data Extraction Suite in VB.NET.

Visit our website to get a free trial version of ByteScout Data Extraction Suite. Free trial contains many of source code samples to help you with your VB.NET project.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Data Extraction Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Data Extraction Suite](#)

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

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

Source Code Files:

```

Imports System.IO
Imports System.Threading
Imports Bytestream.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 Data Extraction Suite Home Page](#)
[Explore ByteScout Data Extraction Suite Documentation](#)
[Explore Samples](#)
[Sign Up for ByteScout Data Extraction 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)

