

pdf to image rendering in background thread with pdf renderer sdk in VB.NET and ByteScout Premium Suite

Simple tutorial on how to do pdf to image rendering in background thread with pdf renderer sdk in VB.NET

This page explains the steps and algorithm of implementing pdf to image rendering in background thread with pdf renderer sdk and how to make it work in your application. ByteScout Premium Suite was created to assist pdf to image rendering in background thread with pdf renderer sdk in VB.NET. ByteScout Premium Suite is the set that includes 12 SDK products from ByteScout including tools and components for PDF, barcodes, spreadsheets, screen video recording.

This rich and prolific sample source code in VB.NET for ByteScout Premium Suite contains various functions and options you should do calling the API to implement pdf to image rendering in background thread with pdf renderer sdk. Follow the steps-by-step instructions from the scratch to work and copy and paste code for VB.NET into your editor. Further improvement of the code will make it more robust.

ByteScout Premium Suite is available as a free trial. You may get it from our website along with all other source code samples for VB.NET applications.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Premium Suite](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout Premium 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 Bytescout.PDFRenderer

Class Program

    Private Shared _doneEvent As New ManualResetEvent(False) ' synchronization event
    Private Shared _counter As Integer ' thread counter

    Friend Shared Sub Main(args As String())

        ' Get all PDF files in current directory
        Dim pdfFiles As String() = Directory.GetFiles(".", "*.pdf")
        _counter = pdfFiles.Length

        ' Render PDF files in separate threads
        For Each pdfFile As String In pdfFiles
            Dim backgroundThread As New Thread(AddressOf BackgroundThreadProc)
            backgroundThread.Start(pdfFile)
            Console.WriteLine(pdfFile & " - conversion started.")
        Next

        ' Wait until threads finished
        _doneEvent.WaitOne()

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

    ' Rendering thread function
    Private Shared Sub BackgroundThreadProc(data As Object)

        Dim fileName As String = DirectCast(data, String)

        Try
            ' Create renderer
            Using renderer As New RasterRenderer()
                renderer.LoadDocumentFromFile(fileName)

                ' Setup rendering
                Dim renderingOptions As New RenderingOptions()
                renderingOptions.JPEGQuality = 90

                Dim renderingResolution As Single = 300

                ' Render document pages
                For i As Integer = 0 To renderer.GetPageCount() - 1
                    renderer.Save(fileName & ".page" & i & ".jpg", RasterImageFormat.JPEG)
                Next
            End Using

            Console.WriteLine(fileName & " - successfully converted.")
        Catch exception As Exception
    End Sub

```

```
        Console.WriteLine(exception.ToString())
    End Try

    If Interlocked.Decrement(_counter) = 0 Then
        ' Set event if all threads finished
        _doneEvent.Set()
    End If
End Sub
End Class
```

VIDEO

<https://www.youtube.com/watch?v=NEwNs2b9YN8>

ON-PREMISE OFFLINE SDK

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