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

Learn pdf to image rendering in background thread with pdf renderer sdk in VB.NET

Here you may get thousands pre-made source code samples for simple implementation in your own programming VB.NET projects. ByteScout PDF Suite was created to assist pdf to image rendering in background thread with pdf renderer sdk in VB.NET. ByteScout PDF Suite is the set that includes 6 SDK products 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.

If you want to quickly learn then these fast application programming interfaces of ByteScout PDF Suite for VB.NET plus the guideline and the VB.NET code below will help you quickly learn pdf to image rendering in background thread with pdf renderer sdk. To use pdf to image rendering in background thread with pdf renderer sdk in your VB.NET project or application just copy & paste the code and then run your app! Further improvement of the code will make it more robust.

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

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

Source Code Files:

Program.vb

```vb
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.J
                Next
            End Using
```

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

visit www.PDF.co