

textextractor progress indication with pdf extractor sdk in VB.NET with ByteScout PDF Suite

Learn textextractor progress indication with pdf extractor sdk in VB.NET

Every ByteScout tool includes sample VB.NET source codes that you can find here or in the folder with installed ByteScout product. ByteScout PDF Suite helps with textextractor progress indication with pdf extractor sdk in VB.NET. 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.

This rich and prolific sample source code in VB.NET for ByteScout PDF Suite contains various functions and options you should do calling the API to implement textextractor progress indication with pdf extractor sdk. To use textextractor progress indication with pdf extractor sdk in your VB.NET project or application just copy & paste the code and then run your app! VB.NET application implementation mostly involves various stages of the software development so even if the functionality works please check it with your data and the production environment.

Our website gives free trial version of ByteScout PDF Suite. It includes all these source code samples with the purpose to assist you with your VB.NET application implementation.

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:

Program.vb

```
Imports Bytescout.PDFExtractor

Module Program

    Sub Main()

        Try
            ' Read all file content...
            Using extractor As TextExtractor = New TextExtractor()
                ' Load document
                extractor.LoadDocumentFromFile("sample.png")

                ' Extractor Progress event
                Console.WriteLine("Text Extraction in progress: " & vbCrLf)
                AddHandler extractor.ProgressChanged, AddressOf Extractor_ProgressChanged

                ' Set option to repair text
                extractor.OCRMode = OCRMode.TextFromImagesAndVectorsAndRepairedFonts

                ' Enable Optical Character Recognition (OCR)
                ' in .Auto mode (SDK automatically checks if needs to use OCR or not)
                extractor.OCRMode = OCRMode.Auto

                ' Set the location of OCR language data files
                extractor.OCRLanguageDataFolder = "c:\Program Files\Bytescout PDF Extrac

                ' Set OCR language
                extractor.OCRLanguage = "eng" ' "eng" for english, "deu" for German, "-
                ' Find more language files at https://github.com/bytescout/ocrdata

                ' Set PDF document rendering resolution
                extractor.OCRResolution = 300

                ' Read all text
                Dim allExtractedText = extractor.GetText()
                Console.WriteLine(vbLf & vbCrLf & "Extracted Text:" & vbCrLf & vbCrLf & "{0}"

            End Using

            Catch ex As Exception
                Console.WriteLine(ex.Message)
            End Try

            Console.WriteLine("Press enter key to exit...")
            Console.ReadLine()

        End Sub

        ''' <summary>
        ''' Handle progress change event
        ''' </summary>
        Private Sub Extractor_ProgressChanged(ByVal sender As Object, ByVal ongoingOperati
```

```

        drawTextProgressBar(Convert.ToInt32(progress), 100)
    End Sub

''' <summary>
''' Display progress bar
''' </summary>
Private Sub drawTextProgressBar(ByVal progress As Integer, ByVal total As Integer)
    'draw empty progress bar
    Console.CursorLeft = 0
    Console.Write("[") ' start
    Console.CursorLeft = 32
    Console.Write("]") ' end
    Console.CursorLeft = 1
    Dim onechunk As Single = 30.0F / total
    Dim position As Integer = 1

    ' draw filled part
    For i As Integer = 0 To onechunk * progress - 1
        Console.BackgroundColor = ConsoleColor.Green
        Console.CursorLeft = Math.Min(System.Threading.Interlocked.Increment(position), 31)
        Console.Write(" ")
    Next

    ' draw unfilled part
    For i As Integer = position To 31
        Console.BackgroundColor = ConsoleColor.Gray
        Console.CursorLeft = Math.Min(System.Threading.Interlocked.Increment(position), 31)
        Console.Write(" ")
    Next

    ' draw totals
    Console.CursorLeft = 35
    Console.BackgroundColor = ConsoleColor.Black
    Console.Write(progress.ToString() & " of " & total.ToString() & " ") ' blank space
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](#)

[visit www.PDF.co](#)

www.bytescout.com