

textextractor progress indication in VB.NET using ByteScout PDF Extractor SDK

How to use ByteScout PDF Extractor SDK for textextractor progress indication in VB.NET

Sample source codes below will show you how to cope with a difficult task, for example, textextractor progress indication in VB.NET. ByteScout PDF Extractor SDK was made to help with textextractor progress indication in VB.NET. ByteScout PDF Extractor SDK is the SDK that helps developers to extract data from unstructured documents, pdf, images, scanned and electronic forms. Includes AI functions like automatic table detection, automatic table extraction and restructuring, text recognition and text restoration from pdf and scanned documents. Includes PDF to CSV, PDF to XML, PDF to JSON, PDF to searchable PDF functions as well as methods for low level data extraction.

You will save a lot of time on writing and testing code as you may just take the code below and use it in your application. In order to implement this functionality, you should copy and paste code below into your app using code editor. Then compile and run your application. This basic programming language sample code for VB.NET will do the whole work for you in implementing textextractor progress indication in your app.

Trial version can be downloaded from our website. Source code samples for VB.NET and documentation are included.

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout PDF Extractor SDK](#)

[Explore API Documentation](#)

[Get Free Training for ByteScout PDF Extractor SDK](#)

[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=s28W3_KMraU

ON-PREMISE OFFLINE SDK

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