

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

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

:

Step-by-step instructions on how to do textextractor progress indication with pdf extractor sdk in VB.NET

These source code samples are assembled by their programming language and functions they use. ByteScout Data Extraction Suite was created to assist textextractor progress indication with pdf extractor sdk in VB.NET. ByteScout Data Extraction Suite is the bundle that includes three SDK tools for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK.

If you want to quickly learn then these fast application programming interfaces of ByteScout Data Extraction Suite for VB.NET plus the guideline and the VB.NET code below will help you quickly learn textextractor progress indication with pdf extractor sdk. Follow the steps-by-step instructions from the scratch to work and copy and paste code for VB.NET into your editor. Want to see how it works with your data then code testing will allow the function to be tested and work properly.

Trial version can be downloaded from our website for free. It contains this and other source code samples for VB.NET.

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:

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(positi
        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(positi
        Console.Write(" ")
    Next

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

www.bytescout.com