

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

Learn to code in VB.NET to make textextractor progress indication with pdf extractor sdk with this simple How-To tutorial

The documentation is crafted to assist you to apply the features on your side easily. ByteScout Premium Suite was created to assist textextractor progress indication with pdf extractor 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.

The below SDK samples describe how to quickly make your application do textextractor progress indication with pdf extractor sdk in VB.NET with the help of ByteScout Premium Suite. 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! Want to see how it works with your data then code testing will allow the function to be tested and work properly.

Our website gives free trial version of ByteScout Premium 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 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 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 ongoingOperatio
            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 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](http://www.bytescout.com)