

How to index PDF files in VB.NET and ByteScout PDF Extractor SDK

How to index PDF files in VB.NET

The coding tutorials are designed to help you test the features without need to write your own code. ByteScout PDF Extractor SDK: the SDK is designed to help developers with pdf tables and pdf data extraction from unstructured documents like pdf, tiff, scans, images, scanned and electronic forms. The library is powered by OCR, computer vision and AI to provide unique functionality like table detection, automatic table structure extraction, data restoration, data restructuring and reconstruction. Supports PDF, TIFF, PNG, JPG images as input and can output CSV, XML, JSON formatted data. Includes full set of utilities like pdf splitter, pdf merger, searchable pdf maker and other utilities. It can index PDF files in VB.NET.

This rich sample source code in VB.NET for ByteScout PDF Extractor SDK includes the number of functions and options you should do calling the API to index PDF files. Just copy and paste the code into your VB.NET application's code and follow the instruction. Code testing will allow the function to be tested and work properly with your data.

Free trial version of ByteScout PDF Extractor SDK is available on our website. Documentation and source code samples are included.

VB.NET - Program.vb

```
Imports System.IO
Imports Bytescout.PDFExtractor

Class Program
    Friend Shared Sub Main(ByVal args As String())

        ' Create Bytescout.PDFExtractor.InfoExtractor instance
        Dim infoExtractor As New InfoExtractor()
        infoExtractor.RegistrationName = "demo"
        infoExtractor.RegistrationKey = "demo"

        ' Create Bytescout.PDFExtractor.TextExtractor instance
        Dim textExtractor As New TextExtractor()
        textExtractor.RegistrationName = "demo"
        textExtractor.RegistrationKey = "demo"

        ' List all PDF files in directory
        For Each file As String In Directory.GetFiles("..\\..\\..\\..", "*.pdf")
            infoExtractor.LoadDocumentFromFile(file)

            Console.WriteLine("File Name:      " & Path.GetFileName(file))
            Console.WriteLine("Page Count:    " & infoExtractor.GetPageCount())
            Console.WriteLine("Author:        " & infoExtractor.Author)
        End For
    End Sub
End Class
```

```
Console.WriteLine("Title:          " & infoExtractor.Title)
Console.WriteLine("Producer:       " & infoExtractor.Producer)
Console.WriteLine("Subject:        " & infoExtractor.Subject)
Console.WriteLine("CreationDate:   " & infoExtractor.CreationDate)
Console.WriteLine("Text (2 lines): ")

textExtractor.LoadDocumentFromFile(file)
Using stringReader As New StringReader(textExtractor.GetTextFromPage(0))
    Console.WriteLine(stringReader.ReadLine())
    Console.WriteLine(stringReader.ReadLine())
End Using
Console.WriteLine()
Next

' Cleanup
infoExtractor.Dispose()
textExtractor.Dispose()

Console.WriteLine()
Console.WriteLine("Press any key to continue...")
Console.ReadLine()
End Sub
End Class
```

FOR MORE INFORMATION AND FREE TRIAL:

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

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

[Explore documentation](#)

[Visit www.ByteScout.com](http://www.ByteScout.com)

or

[Get Your Free API Key for www.PDF.co Web API](#)