How to convert protected PDF document to XLS in VB.NET and ByteScout PDF Extractor SDK

The tutorial shows how to convert protected PDF document to XLS in VB.NET

These source code samples are listed and grouped by their programming language and functions they use. Want to convert protected PDF document to XLS in your VB.NET app? ByteScout PDF Extractor SDK is designed for it. 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.

Fast application programming interfaces of ByteScout PDF Extractor SDK for VB.NET plus the instruction and the code below will help you quickly learn how to convert protected PDF document to XLS. This VB.NET sample code is all you need for your app. Just copy and paste the code, add references (if needs to) and you are all set! Enjoy writing a code with ready-to-use sample codes in VB.NET.

ByteScout free trial version is available for download from our website. It includes all these programming tutorials along with source code samples.

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

Source Code Files:

Program.vb

```
Imports System.IO
Imports Bytescout.PDFExtractor
Imports System.Diagnostics

Class Program

    Friend Shared Sub Main(args As String())

        ' Create Bytescout.PDFExtractor.XLSExtractor instance

        Dim extractor As New XLSExtractor()
        extractor.RegistrationName = "demo"
        extractor.RegistrationKey = "demo"

        File.Delete("output.xls")

        ''' Document Password Can be set in two ways
        ''' 1. Using Property
        ''' 2. Using Events

        ' 1. Handle document password using Property
        ' extractor.Password = "password"

        ' 2. Handle document password using Event
        AddHandler extractor.PasswordRequired, New System.EventHandler(AddressOf extrac

        ' Load sample PDF document
        extractor.LoadDocumentFromFile("encrypted (password is 'password').pdf")

        ' Uncomment this line if you need all pages converted into a single worksheet:
        'extractor.PageToWorksheet = False

        ' Set the output format to XLS
        extractor.OutputFormat = SpreadseetOutputFormat.XLS

        ' Save the spreadsheet to file
        extractor.SaveToXLSFile("output.xls")

        ' Cleanup
        extractor.Dispose()

        ' Open result file in default associated application (for demo purposes)
        Process.Start("output.xls")

    End Sub

    Private Shared Sub extractor_PasswordRequired(sender As Object, e As EventArgs)

        ' Ask user for password and put it to `Password` property.
        CType(sender, XLSExtractor).Password = "password"

    End Sub
End Class
```

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

visit www.PDF.co