How to convert protected pdf document to xls with pdf extractor sdk in VB.NET and ByteScout Data Extraction Suite

Learn to convert protected pdf document to xls with pdf extractor sdk in VB.NET

The code displayed below will guide you to install an VB.NET app to convert protected pdf document to xls with pdf extractor sdk. Want to convert protected pdf document to xls with pdf extractor sdk in your VB.NET app? ByteScout Data Extraction Suite is designed for it. ByteScout Data Extraction Suite is the set that includes 3 SDK products for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK.

Want to save time? You will save a lot of time on writing and testing code as you may just take the VB.NET code from ByteScout Data Extraction Suite for convert protected pdf document to xls with pdf extractor sdk below and use it in your application. 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! This basic programming language sample code for VB.NET will do the whole work for you to convert protected pdf document to xls with pdf extractor sdk.

All these programming tutorials along with source code samples and ByteScout free trial version are available for download from our website.

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

Source Code Files:

```
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 extra
        ' 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")
        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
```

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

visit www.PDF.co

www.bytescout.com