scanned PDF to text in VB6 using ByteScout PDF Extractor SDK

How To: tutorial on scanned PDF to text in VB6

Today you are going to learn how to scanned PDF to text in VB6. ByteScout PDF Extractor SDK was made to help with scanned PDF to text in VB6. 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.

This rich sample source code in VB6 for ByteScout PDF Extractor SDK includes the number of functions and options you should do calling the API to implement scanned PDF to text. Follow the instruction from the scratch to work and copy and paste code for VB6 into your editor. This basic programming language sample code for VB6 will do the whole work for you in implementing scanned PDF to text in your app.

Free trial version of ByteScout PDF Extractor SDK is available on our website. Get it to try other samples for VB6.

VB6 - Form1.frm

```
VERSION 5.00
Begin VB.Form Form1
   Caption         =   "Scanned PDF to Text"
   ClientHeight    =   1095
   ClientLeft      =   120
   ClientTop       =   465
   ClientWidth     =   3675
   LinkTopic       =   "Form1"
   ScaleHeight     =   1095
   ScaleWidth      =   3675
   StartUpPosition =   3  'Windows Default
   Begin VB.CommandButton cmd_scanned_pdf_to_text
      Caption         =   "Convert Scanned PDF to Text"
      Height          =   855
      Left            =   120
      TabIndex        =   0
      Top             =   120
      Width           =   3495
   End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False
Private Sub cmd_scanned_pdf_to_text_Click()

    ' Hanlde Error
    On Error GoTo ErrorHandler:

    ' Create Bytescout.PDFExtractor.TextExtractor object
    Set extractor = CreateObject("Bytescout.PDFExtractor.TextExtractor")

    ' Set Registration name and key
    extractor.RegistrationName = "demo"
    extractor.RegistrationKey = "demo"

    ' Load sample PDF document
    extractor.LoadDocumentFromFile "sample.pdf"

    ' Enable Optical Character Recognition (OCR)
    extractor.OCRMode = 1 ' OCRMode.Auto = 1

    ' Set the location of OCR language data files
    extractor.OCRLanguageDataFolder = "c:\Program Files\Bytescout PDF Extractor
SDK\ocrdata"

    ' Set OCR language
    ' "eng" for english, "deu" for German, "fra" for French, "spa" for Spanish etc -
according to files in "ocrdata" folder.
    extractor.OCRLanguage = "eng"
    ' Find more language files at https://github.com/bytescout/ocrdata

    ' Set PDF document rendering resolution
    extractor.OCRResolution = 300

    ' You can also apply various preprocessing filters to improve the recognition on
low-quality scans.
    ' But they significantly hit the performance, so do not enable them by default.

    ' Automatically deskew skewed scans
    'extractor.OCRImagePreprocessingFilters.AddDeskew()

    ' Remove vertical or horizontal lines (sometimes helps to avoid OCR engine's page
segmentation errors)
    'extractor.OCRImagePreprocessingFilters.AddVerticalLinesRemover()
    'extractor.OCRImagePreprocessingFilters.AddHorizontalLinesRemover()

    ' Repair broken letters
    'extractor.OCRImagePreprocessingFilters.AddDilate()

    ' Remove noise
    'extractor.OCRImagePreprocessingFilters.AddMedian()

    ' Apply Gamma Correction
    'extractor.OCRImagePreprocessingFilters.AddGammaCorrection()

    ' Add Contrast
    ' extractor.OCRImagePreprocessingFilters.AddContrast(20)


    ' (!) You can use new OCRAnalyzer class to find an optimal set of image
preprocessing
    ' filters for your specific document.
```

```
    ' See "OCR Analyser" example.


    ' Peform Save to Text file
    extractor.SaveTextToFile "output.txt"

    ' Show Success Message
    MsgBox "Extracted data from scanned PDF are saved to 'output.text' file.",
vbInformation, "Success"

    ' Close form
    Unload Me

ErrorHandler:
If Err.Number <> 0 Then
    MsgBox Err.Description, vbInformation, "Error"
End If

End Sub
```

VB6 - Scanned_PDF_To_Text.vbp

```
Type=Exe
Reference=*\G{00020430-0000-0000-C000-
000000000046}#2.0#0#..\..\..\..\..\..\..\..\..\Windows\SysWOW64\stdole2.tlb#OLE
Automation
Reference=*\G{F1D62CEE-68AA-4F38-9DB0-
8021C13255D8}#9.1#0#..\..\..\..\..\..\..\..\..\WINDOWS\SYSWOW64\Bytescout.PDFRenderer.t
 PDF Renderer SDK [TRIAL]
Form=Form1.frm
Startup="Form1"
Command32=""
Name="ScannedPDFToText"
HelpContextID="0"
CompatibleMode="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionCompanyName="Hiren"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FlPointCheck=0
FDIVCheck=0
```

```
UnroundedFP=0
StartMode=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=1
```

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

or

Get Your Free API Key for www.PDF.co Web API