

How to extract text from areas in VBScript and VB6 using ByteScout Text Recognition SDK

This tutorial will show how to extract text from areas in VBScript and VB6

These source code samples are listed and grouped by their programming language and functions they use. Want to extract text from areas in your VBScript and VB6 app? ByteScout Text Recognition SDK is designed for it. ByteScout Text Recognition SDK is the text recognition SDK to help with extraction of text using OCR from scanned images and documents. Supports English and non-Latin languages, can take PDF as input.

This code snippet below for ByteScout Text Recognition SDK works best when you need to quickly extract text from areas in your VBScript and VB6 application. Just copy and paste the code into your VBScript and VB6 application's code and follow the instruction. You can use these VBScript and VB6 sample examples in one or many applications.

You can download free trial version of ByteScout Text Recognition SDK from our website to see and try many others source code samples for VBScript and VB6.

VBScript and VB6 - ExtractFromAreas.vbs

```
' Create and activate TextRecognizer object
Set textRecognizer = CreateObject("ByteScout.TextRecognition.TextRecognizer")
textRecognizer.RegistrationName = "demo"
textRecognizer.RegistrationKey = "demo"

Set comHelpers = textRecognizer.ComHelpers

inputDocument = "..\..\areas-sample.pdf"
pageIndex = 0
outputDocument = "result.txt"

' Load document (image or PDF)
textRecognizer.LoadDocument(inputDocument)

' Set the location of OCR language data files
textRecognizer.OCRLanguageDataFolder = "c:\Program Files\ByteScout Text Recognition
SDK\ocrdata_best\"

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

```
' Get page size (in pixels). Size of PDF document is computed from PDF Points
' and the rendering resolution specified by `textRecognizer.PDFRenderingResolution`
(default 300 DPI)
Dim pageWidth, pageHeight
pageWidth = textRecognizer.GetPageWidth(pageIndex)
pageHeight = textRecognizer.GetPageHeight(pageIndex)

' Add area of interest as a rectangle at the top-right corner of the page
textRecognizer.RecognitionAreas.Add pageWidth / 2, 0, pageWidth / 2, 300
' Add area of interest as a rectangle at the bottom-left corner of the page,
' and indicate it should be rotated at 90 deg
textRecognizer.RecognitionAreas.Add 0, pageHeight / 2, 300, pageHeight / 2,
comHelpers.AreaRotation_Rotate90FlipNone

' Now you can get recognized text for further analysis as a list of objects
' containing coordinates, object kind, confidence.
Set ocrObjectList = textRecognizer.GetOCRObjects(pageIndex)
For Each ocrObject in OCRObjectList
    WScript.Echo ocrObject.Text & " [" & ocrObject.X & ", " & ocrObject.Y & ", " &
ocrObject.Width & ", " & ocrObject.Height & "]" : " & ocrObject.Confidence
Next

' ... or you can save recognized text pieces to file
textRecognizer.KeepTextFormatting = False ' save without formatting
textRecognizer.SaveText outputDocument, pageIndex, pageIndex

WScript.Echo "Extracted text saved to " + outputDocument

Set recognizer = Nothing
```

FOR MORE INFORMATION AND FREE TRIAL:

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

[Read more about ByteScout Text Recognition SDK](#)

[Explore documentation](#)

[Visit www.ByteScout.com](#)

or

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