www.bytescout.com

barcode image preprocessing filters with barcode reader sdk in VBScript and ByteScout Data Extraction Suite

barcode image preprocessing filters with barcode reader sdk in VBScript

Easy to understand coding instructions are written to assist you to try-out the features without the requirement to write your own code. ByteScout Data Extraction Suite was made to help with barcode image preprocessing filters with barcode reader sdk in VBScript. ByteScout Data Extraction Suite is the bundle that includes three SDK tools for data extraction from PDF, scans, images and from spreadsheets: PDF Extractor SDK, Data Extraction SDK, Barcode Reader SDK.

If you want to quickly learn then these fast application programming interfaces of ByteScout Data Extraction Suite for VBScript plus the guideline and the VBScript code below will help you quickly learn barcode image preprocessing filters with barcode reader sdk. Follow the steps-by-step instructions from the scratch to work and copy and paste code for VBScript into your editor. Want to see how it works with your data then code testing will allow the function to be tested and work properly.

Trial version along with the source code samples for VBScript can be downloaded 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:

TestBarcodeReading.vbs

```
' This exmaple demonstrates the use of image filters to improve the decoding or speed.

Dim result

Set reader = CreateObject("Bytescout.BarCodeReader.Reader")
reader.RegistrationName = "demo"
reader.RegistrationKey = "demo"

' Set barcode type to find
reader.BarcodeTypesToFind.Code128 = True


' WORKING WITH LOW CONTRAST BARCODE IMAGES

' Add the contrast adjustment for the low contrast image
reader.ImagePreprocessingFilters.AddContrast(40)

result = result & "Image ""low-contrast-barcode.png""" & vbCRLF

reader.ReadFromFile "low-contrast-barcode.png"

If reader.FoundCount = 0 Then
        result = result & "No barcode found!" & vbCRLF
Else
        For i = 0 To reader.FoundCount - 1
        result = result & "Found barcode with type " & CStr(reader.GetFoundBarcodeType(
    Next
End If

reader.ImagePreprocessingFilters.Clear()
result = result & vbCRLF


' WORKING WITH NOISY BARCODE IMAGES

' Add the median filter to lower the noise
reader.ImagePreprocessingFilters.AddMedian()

result = result & "Image ""noisy-barcode.png""" & vbCRLF

reader.ReadFromFile "noisy-barcode.png"

If reader.FoundCount = 0 Then
        result = result & "No barcode found!" & vbCRLF
Else
        For i = 0 To reader.FoundCount - 1
        result = result & "Found barcode with type " & CStr(reader.GetFoundBarcodeType(
    Next
End If

reader.ImagePreprocessingFilters.Clear()
result = result & vbCRLF


' WORKING WITH DENSE AND ILLEGIBLE BARCODES
```

```
' Add the scale filter to enlarge the barcode to make gaps between bars more distinguis
reader.ImagePreprocessingFilters.AddScale_2(2) ' enlarge twice

result = result & "Image ""too-dense-barcode.png""" & vbCRLF

reader.ReadFromFile "too-dense-barcode.png"

If reader.FoundCount = 0 Then
        result = result & "No barcode found!" & vbCRLF
Else
        For i = 0 To reader.FoundCount - 1
        result = result & "Found barcode with type " & CStr(reader.GetFoundBarcodeType(
    Next
End If

reader.ImagePreprocessingFilters.Clear()
result = result & vbCRLF


' REMOVE EMPTY MARGINS FROM IMAGE TO SPEED UP THE PROCESSING

' Add the crop filter to cut off empty margins from the image.
' This will not improve the recognition quality but may speed up the processing
' if you enabled multiple barcode types to search.
reader.ImagePreprocessingFilters.AddCropDark()

result = result & "Image ""barcode-with-large-margins.png""" & vbCRLF

reader.ReadFromFile "barcode-with-large-margins.png"

If reader.FoundCount = 0 Then
        result = result & "No barcode found!" & vbCRLF
Else
        For i = 0 To reader.FoundCount - 1
        result = result & "Found barcode with type " & CStr(reader.GetFoundBarcodeType(
    Next
End If

reader.ImagePreprocessingFilters.Clear()
result = result & vbCRLF


Msgbox result


Set reader = Nothing
```

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