

How to use image preprocessing filters in VB.NET with ByteScout Text Recognition SDK

How to use image preprocessing filters in VB.NET

These sample source codes on this page below are demonstrating how to use image preprocessing filters in VB.NET. What is ByteScout Text Recognition SDK? It 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. It can help you to use image preprocessing filters in your VB.NET application.

You will save a lot of time on writing and testing code as you may just take the VB.NET code from ByteScout Text Recognition SDK for use image preprocessing filters below and use it in your application. In your VB.NET project or application you may simply copy & paste the code and then run your app! Enjoy writing a code with ready-to-use sample codes in VB.NET.

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

VB.NET - Module1.vb

```
Imports Bytescout.TextRecognition

Module Module1

    Sub Main()

        Dim inputDocument As String = ".\skewed.png"
        Dim outputDocument As String = ".\result.txt"

        ' Create and activate TextRecognizer instance
        Using textRecognizer As TextRecognizer = New TextRecognizer("demo", "demo")

            Try
                ' 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"
            End Try
        End Using
    End Sub
End Module
```

```

        ' Add deskew filter that automatically rotates the image to make the
text horizontal.
        ' Note, it analyzes the left edge of scanned text. Any dark artifacts
may prevent
        ' the correct angle detection.
textRecognizer.ImagePreprocessingFilters.AddDeskew()

        ' Other filters that may be useful to improve recognition
        ' (note, the filters are applied in the order they were added):

        ' Improve image contrast.
'textRecognizer.ImagePreprocessingFilters.AddContrast()

        ' Apply gamma correction.
'textRecognizer.ImagePreprocessingFilters.AddGammaCorrection()

        ' Apply median filter. Helps to remove noise.
'textRecognizer.ImagePreprocessingFilters.AddMedian()

        ' Apply dilate filter. Helps to cure symbols erosion.
'textRecognizer.ImagePreprocessingFilters.AddDilate()

        ' Lines removers. Removing borders of some tables may improve the
recognition.
'textRecognizer.ImagePreprocessingFilters.AddHorizontalLinesRemover()
'textRecognizer.ImagePreprocessingFilters.AddVerticalLinesRemover()

        ' Recognize text from all pages and save it to file
textRecognizer.SaveText(outputDocument)

        ' Open the result file in default associated application (for demo
purposes)
        Process.Start(outputDocument)

        Catch exception As Exception

        Console.WriteLine(exception)

    End Try

End Using

End Sub

End Module

```

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

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

[Explore documentation](#)

[Visit \[www.ByteScout.com\]\(http://www.ByteScout.com\)](#)

or

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