

Documentation

INBarcode ActiveX/COM

V1.7.1



Inspirant
Volker Schmid
Roemerstrasse 39
D-78183 Huefingen
Germany

Table Of Content

Introduction.....	3
Licence.....	4
ActiveX wrapper.....	5
Usage.....	5
Installation.....	6
ActiveX functions.....	7
Finding and decoding barcodes.....	7
Decoding given single barcodes.....	8
Retrieving results.....	9
Other functions.....	11
Disable features.....	13
Recognition Problems?.....	14

Introduction

Using the INBarcodeOCR you are able to recognize and decode barcodes in an image.

Features:

- recognizes **Code39** (with optional calculation an revision of checksum)
- recognizes **Code128** (including calculation an revision of checksum)
- recognizes **EAN128** (including calculation an revision of checksum)
- recognizes **EAN13** (including calculation an revision of checksum)
- recognizes **EAN8** (including calculation an revision of checksum)
- recognizes **UPC-A** (including calculation an revision of checksum)
- recognizes **2/5** and **2/5 interleaved**
- able to find all supported barcodes inside documents
- supports **BMP**, **TIF**, **JPG** and **PNG** image formats
- supports **PDF** file format
- can **directly use handle** of existing bitmaps in memory (bitmal handle hDC)
- returns position, type, orientation and content of every supported barcode found
- methods to decode single barcode images
- recognizes barcodes rotated by 180°, horizontal and vertical
- runs on W98/ME/2000/2000 Server/XP/2003 Server

Restrictions:

- the maximum allowed aberration from horizontal or vertical is +6° / -6°
- every given bitmap must be smaller 8192x8192 pixel
- images must be at least 200DPI
- depending on the size and quality of the barcodes, 200 to 400 DPI is recommended
- if you use multipage TIF files, the library uses only the first page

Special ActiveX/COM restrictions:

- The wrapper can not handle the following functions of the DLL:
 - PDFSetResolution()
 - PDFEnabled()
 - PDFGetImage()
 - PDFImageToPDF()
- Though, the wrapper is able to handle PDF images using FindBarcodesFile()

Licence

Please refer the DLL documentation for actual licensing information.

ActiveX wrapper

Usage

This is the documentation for the ActiveX wrapper to the INBarcodeOCR.DLL. The ActiveX wrapper allows you the integration in VBScript, ASP and other languages supporting ActiveX or COM objects.

Here is an example usage in VBScript:

```
strFilename = "c:\barcode filename.tif"

' create a new object of type INBarcodeCOM
Set objINBarcode = CreateObject("INBarcodeCOM.INBarcode")

' read barcodes
lngCodeCount = objINBarcode.FindBarcodesFile(strFilename)

strBarcode = "Barcodes: " & lngCodeCount & vbCrLf

' get results
For x = 1 To lngCodeCount
    strBarcode = strBarcode & "Code: " & _
        objINBarcode.GetBarcodeCode(x) & vbCrLf
    strBarcode = strBarcode & " Type: " & _
        objINBarcode.GetBarcodeType(x) & vbCrLf
    strBarcode = strBarcode & " Orientation: " & _
        objINBarcode.GetBarcodeOrientation(x) & vbCrLf
    strBarcode = strBarcode & vbCrLf
Next

Set objINBarcode = Nothing ' clean up

MsgBox strBarcode
```

The example above processes a given image, opens a messagebox and shows the number and content of the recognized barcodes.

NOTE:

The example above is also working in VB6 without any changes except of declaring the variables.

Installation

To **install** the INBarcodeCOM wrapper, please copy the INBarcodeCOM.dll to the System path of windows (c:\windows\system32\ or c:\winnt\system32\). This is the same path where you should install the INBarcodeOCR.DLL, too.

To **add the PDF support**, the INBarcodePDF.dll has to be installed into the same directory than the INBarcodeOCR.dll. If you don't need the PDF support, you can omit the INBarcodePDF.dll.

To **register** the component, please register the INBarcodeCOM.dll once using regsvr32. This may be automated using your installer („register component“).

The INBarcodeCOM.exe needs the installed **VB6 runtime files** on the computer. There is a runtime setup available from Microsoft (link: <http://support.microsoft.com/kb/290887/EN-US/>). It can be started using the /Q option to perform a quiet installation in your setups.

ActiveX functions

Finding and decoding barcodes

This routines are meant to find and recognize barcodes in a document image. Use this functions if you don't have an isolated barcode image. The complete image is scanned for possible barcodes and then everyone of this candidates is checked against the possible barcode-formats.



FindBarcodesFile (Filename)

Retrieves the barcodes in a given image-file (BMP, JPG, PNG, TIF, PDF). Returns the number of barcodes found.

Returns -1 for an error (file not found)

FindBarcodesClipboard()

Retrieves the barcodes in a given image in clipboard. Returns the number of barcodes found.

Returns -1 for an error (no image in clipboard)

FindBarcodesHDC (lngHandle)

Retrieves the barcodes in a given image. Provide an existing bitmap-handle (hDC) as parameter to define the image to use. This may be a very good way to process images that are already loaded by your application. Don't forget to release your hDC after processing inside your application.

Return values:

Returns the number of barcodes found.

Returns -1 for an error (long)

Decoding given single barcodes

Use this routines if you have images of isolated barcodes and you don't have the need for finding barcodes in a document.



DecodeBarcodeFile (Filename)

Decodes the barcode in a given image-file (BMP, JPG, PNG, TIF). Returns the number of barcodes found (always 0 or 1).

Info: This is not working for PDF-files! Use FindBarcodesFile() instead.

Returns -1 for an error (file not found)

DecodeBarcodeClipboard()

Decodes the barcode of the image in the clipboard. Returns the number of barcodes found (always 0 or 1).

Returns -1 for an error (no image in clipboard)

Retrieving results

This functions are for retrieving the results of a foregoing barcode-recognition. Please call this functions to receive the results of GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard().

GetBarcodeType (BarcodeNumber)

Returns the type of a barcode. The BarcodeNumber ranges from 1 to the number of barcodes found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
Codetype	The type of this recognized barcode. Possible values are: CODE39 CODE128 EAN13 EAN8 EAN128 UPC-A 2/5i 2/5

GetBarcodeCode (BarcodeNumber)

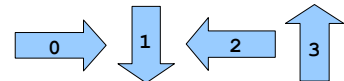
Returns the code of a barcode. The BarcodeNumber ranges from 1 to the number of barcodes found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
Code	The recognized barcode-value.

GetBarcodeOrientation (BarcodeNumber)

Returns the orientation of a barcode. The BarcodeNumber ranges from 1 to the number of barcodes found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
Orientation	The orientation of the barcode. 0 = horizontal 2 = horizontal / rotated 1 = vertical 3 = vertical / rotated



GetBarcodeLeft (BarcodeNumber)

Returns the X coordinate (pixel) of the upper left position of a barcode. The BarcodeNumber ranges from 1 to the number of barcodes found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
X	The upper left position of the rectangle used to identify this barcode.

GetBarcodePage (BarcodeNumber)

Returns the page number, on which the barcode has been found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
PageNumber	The page, the barcode has been found (PDF only)

GetBarcodeTop (BarcodeNumber)

Returns the Y coordinate (pixel) of the upper left position of a barcode. The BarcodeNumber ranges from 1 to the number of barcodes found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
Y	The upper left position of the rectangle used to identify this barcode.

GetBarcodeWidth (BarcodeNumber)

Returns the width (pixel) of a barcode. The BarcodeNumber ranges from 1 to the number of barcodes found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
Width	The width of the rectangle used to identify this barcode.

GetBarcodeHeight (BarcodeNumber)

Returns the height (pixel) of a barcode. The BarcodeNumber ranges from 1 to the number of barcodes found. The number of barcodes found is returned by the GetBarcodesFile(), GetbarcodesClipboard(), DecodeBarcodeFile() and DecodeBarcodeClipboard() functions.

Return-Value	Description
Height	The height of the rectangle used to identify this barcode.

NOTE :

The coordinates returned by this functions (Left, Top, Width, Height) are the coordinates of the image used to recognize the barcode. There is no guarantee that this rectangle covers the complete barcode! In many cases this rectangle covers only a part of the complete barcode. This is meant only for information to the developer in case he wishes to decide something based on the position.

Other functions

GetBarcodeVersionInfo()

Returns the version of the INBarcodeOCR.DLL.

RegisterBarcodeDLL(Name, Password)

Registers this DLL using a name and a password you got after purchase. The *name* parameter is case-sensitive!

Returns *True* for success

Returns *False* for failure

You have to register the INBarcodeCOM component each time you created a new instance to be able to use it. You will receive the name and password after successfully purchasing a licence. Without registering, this library will show a evaluation-reminder at some times calling the FindBarcodesFile() and FindBarcodesClipboard() function.

UseCode39Checksum(Flag)

Decide if the OCR should calculate and revise a checksum for code39 barcodes. Code39 defines no checksum by default but there is an option to calculate and encode this checksum. If your barcodes are Code39 and having a checksum you can enable this checkup with this function.

False = don't calculate and revise checksum (default)

True = calculate and revise

UseIncreasedSensitivity(Flag)

Use increased sensitivity for reading barcodes in colored images.

False = don't use the increased sensitivity (default)

True = use increased sensitivity on colored images

Important: Enabling the increased sensitivity will slowdown recognition noticeable but can produce much more accurate results in color-scanned images with more that 8 bit color-depth.

UseFineSearch(Flag)

Use this option with images less 150 DPI. Such images are often produced using direct barcode drawing instead of scanning.

False = don't use fine search technology (default)

True = use fine search technology

Important: Enabling the fine search technology will slowdown recognition.

ReturnAllCandidates (Flag)

This option allows you to return barcode candidates, too.

False = don't return all candidates (default)

True = return all candidates

Disable features

This section describes functions for disabling some features of the library. The reason may be speed improvements. Another reason may be the wish to receive only one barcode-type even though other barcodes exist on the documents. Especially the abandonment of checking for rotated barcodes may speed up things dramatically.

DisableEAN13 (Flag)

Set if the OCR should search and decode EAN13 and UPC-A barcodes

False = Enable this barcode-type (default)

True = Disable this barcode-type

DisableEAN8 (Flag)

Set if the OCR should search and decode EAN8 barcodes

False = Enable this barcode-type (default)

True = Disable this barcode-type

DisableCode39 (Flag)

Set if the OCR should search and decode Code39 barcodes

False = Enable this barcode-type (default)

True = Disable this barcode-type

DisableCode128 (Flag)

Set if the OCR should search and decode Code128 barcodes

False = Enable this barcode-type (default)

True = Disable this barcode-type

Disable2of5 (Flag)

Set if the OCR should search and decode 2/5 interleaved barcodes

False = Enable this barcode-type (default)

True = Disable this barcode-type

DisableRotation (Flag)

Set if the OCR should search and decode rotated barcodes (by 90°, 180° and 270°)

False = Enable this feature (default)

True = Disable this feature (only left to right barcodes can be found)

Recognition Problems?

Most problems occur when

- barcode-type is not supported by this library
- image quality is bad
- image resolution is too low (<200 DPI)
- less contrast (background colour?)
- barcodes are rotated more than 6°
- barcode is distorted
- the left or right whitespace is too small

Tips:

- try scanning using 300 DPI or even more
- experiment using brightness and contrast of the images (scanner interface)
- try B/W scanning with various thresholds
- remember, that only the first page is scanned inside of TIF and PDF documents

Please refer the INBarcodeOCR DLL documentation for further help and information.