Image Scholar
OCR Recognizer Library
Manual

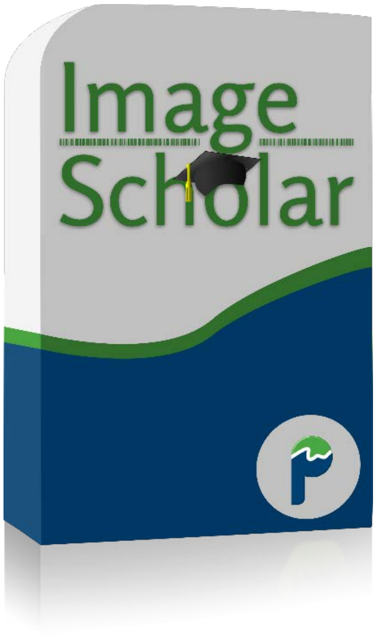


Table of Contents

Introduction

Thank you for choosing Pertech Industries Image Scholar library. The purpose of this library is to provide a low cost, high level, programming interface for applications that need various image recognition algorithms including MICR OCR, Standard Font OCR, Barcode Reading and Micro Hole reading. The Image Scholar library works with images from 200 – 600 dpi and of grey scale 8 bit bmp files. If you need this library customized us the request a quote form at www.perteindustries.com\software.

This manual also include instructions for using the example.exe file that came with the your OCR library package.

Preface

This manual has been written to help you install and use the Hocr.dll library file. Feel free to contact Pertech Industries at 1-307-857-4821 should you need further assistance. Pertech Industries provides technical support Monday through Friday from 8 AM to 4 PM MST.

Legal Notices

Disclaimer

Information in this document is subject to change without notice. Consult the Pertech Industries web site for information that is applicable and current. Pertech Industries reserves the right to improve products as new technology, components, software, and firmware become available.

Pertech Industries will not be liable for personal injury or property damage except for personal injury or property damage caused by Pertech Industries negligence or willful act. In no event shall Pertech Industries be liable for loss of profit, loss of use, loss of revenue, business interruption, direct, consequential or special damages, or for loss or damage caused by or resulting from errors or defects in the graphic materials generated by the equipment furnished hereunder whether such liability is in warranty, negligence, or otherwise.

Copyright

Trademarks

Getting Started

It is a good idea to run the Example.exe file using images already obtained to get a feel for how the Hocr.dll recognizer works. This may also indicate if the Recognizer is appropriate for your needs. To run the Example.exe file see Example.exe section below.

When you are ready to start programming install the OCR Recognizer Library and accompanying files to the working directory and add to the project. Please refer to the Program flow section below.

OCR Engine
Re-image the OCR engine:
Required files:

1. Reimage.dll (to be placed in the working directory or in the system directory.)
2. E13b.rec (to be placed in the working directory)
3. CMC7.rec (to be placed in the working directory)
4. Tft.rec (to be placed in the working directory)

Program Flow

Typically the application will:

1. Load the desired recognizer engine type referencing the type, i.e. TYPE_BARCODE.
   LoadRecognizer(UINT Type);

2. Aquire the image in a Dib (Device Independent Bitmap) alocated memory.

3. Call to perform the recognition. The call requires passing the Dib pointer, passing starting cordianates the size of the recognition window, the type of recogintion for the engine, the direction to read and a pointer to a char array for the result, and the size of the array.
   e.g. MakeOCR(*MyDib, 10, 10, 300, 100, UINT Type, LEFT_RIGHT, *Result, 20);

4. Repeat steps 2 and 3 as needed.
5. Manage the Dib memory appropriately.
6. Unload the Recognizer, **UnloadAllRecognizers(void);**

Example:

```
LoadRecognizer(TYPE_BARCODE);
.......Perform scan an load image to memory ..........

char [20] = Result;
int = CharsRecognized = 0;
CharsRecognized = MakeOCR( *MyDib, 10, 10, 300, 100, UINT Type, LEFT_RIGHT, *Result,
20);

if ...........do something with the recognized data .............

UnloadAllRecognizers(void);
```

All the library calls are described in the Library Functions section .

Library Functions

The OCR Recognizer Library has several configurable options to optimize recognition. The configuration options are describe at the bottom of this section.

BOOL LoadRecognizer (UINT Type, char *CompanyName, char *AuthCode)

There are several unique recognizer types. Only one type can be loaded at a time.
UNIT Type: One of the types listed below to be recognized.

TYPE_NUMERIC	1	// numeric + punctuation
TYPE_CMC7	3	// CMC7
TYPE_E13B	4	// E13B
TYPE_OMNIFONT	5	// omnifont
TYPE_PDF417	6	// PDF417
TYPE_BARCODE	7	// Barcode (2/5, 3/9 ecc.)
TYPE_DATAMATRIX	8	// Datamatrix 2D barcode
TYPE_MICROHOLES	10	// Micro Hole reading

This function returns TRUE if loaded OK or FALSE if an error occurred.

BOOL UnLoadAllRecognizers (Void)

Free the memory from loaded recognizer. MANDATORY to call before closing application or when switching recognizers.
This function returns TRUE.

int MakeOCR_P (char *DibPtr, int x, int y, int width, int height, UINT Type, UINT Direction, char *Result, int ResultSize)

char *DibPtr a reference to the pointer.

int x: The right most starting position, in tenths of a millimeter, of the recognition area with in the image. The recognizer calculates the distance based on the image dpi.

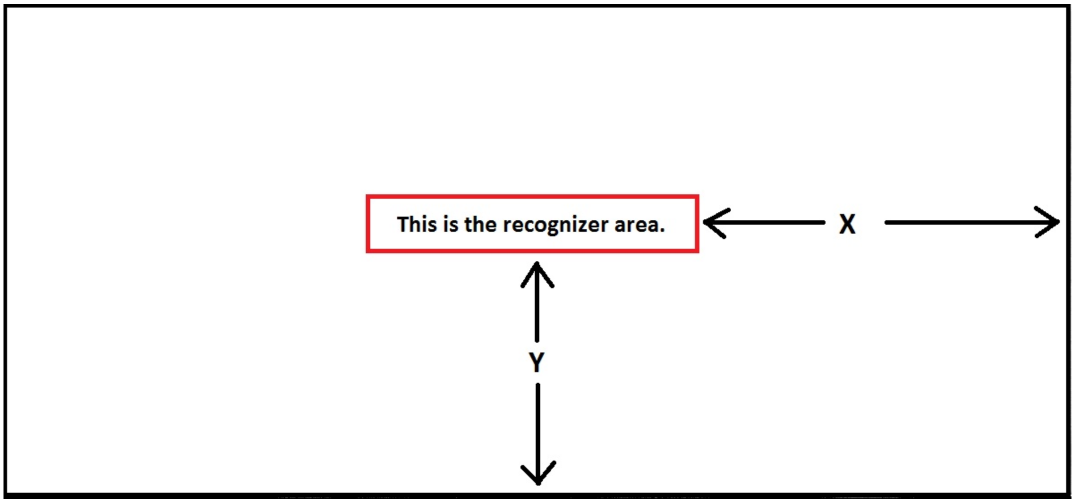
int y: The bottom most starting position, in tenths of a millimeter, of the recognition area with in the image. The recognizer calculates the distance based on the image dpi.

int width: The left to right size, in tenths of a millimeter, of the recognition area with in the image. The recognizer calculates the distance based on the image dpi.

int height: The bottom to top size, in tenths of a millimeter, of the recognition area with in the image. The recognizer calculates the distance based on the image dpi.

Note:

1. *For character recognition the selected area should only include a single line of text.*
2. *The direction is relative to the direction of the original scan See Appendix A.*



UINT TYPE: See LoadRecognizer above.

UINT Direction: The direction to read, there are four directions,

LEFT_RIGHT	0
RIGHT_LEFT	1
UP_DOWN	2

DOWN_UP	3

*char *Result*: A pointer to the char array where the data recognized is placed.

int Resultsize: The number of characters expected or required to load the array with recognized characters.

This function returns an int equal to the number of character recognized.

char * GetRecoDllVersion (void)

Returns a string with the information about the revision of the dll.
char StrVersionInfo[200].

BOOL ChangeSetup (long ChangeType, long val)

Allows using a debug mode, changing the character display when a character is unrecognized and the ability to change the threshold.
long Type: One of the three changeable abilities.

long val: Applicable values for each given change.

CHANGE_DBG_MODE	1	
DBG_NO	0	// Clear all debug mode
DBG_SHOWDIB	1	// Show the recognize Dib
CHANGE_UNREC_CHAR	2	// Character to indicate an unrecognized char. Default *
32 – 256		// Printable character to represent unrecognized characters.
CHANGE_THRESHOLD	3	// Gives a developer fine tuning for their app.
0 – 200		// An accuracy acceptance level, 0 = auto

This function returns TRUE if successful or FALSE if an error occurred.

int HOcrLimitCharset (const char* char_set)

Limit the recognition a given range of characters. Range must be less than
Call this function just before the call to the LoadRecognizer.
Calls to MakeOCRxx will use character set loaded in the last call to HOcrLimitCharset.
If an empty string is entered entire character set will be used.
Valid only for Type = TYPE_OMNIFONT.

This function returns 1 if successful or 0 if an error occurred.

int HOcrConfigBarcode (const char *cfg)

Allows specific configuration of 1D barcode reading.
Call this function after LoadRecognizer.
cfg = “300000000” // to read code 2/5 with check digit only

cfg = “111111111” // to read all barcodes without check digit

Note: default is disabled for all barcodes.

barcode configuration (each char can be a digit - 0 to 9)

char 0 -> code 2/5 interleaved conversion

0 = disable

1 = enable without check digit

3 = enable with check digit

char 1 -> code 3/9 conversion

0 = disable

1 = enable without check digit

3 = enable with check digit

char 2 -> code 128 conversion

0 = disable

1 = enable without check digit

char 3 -> code 93 conversion

0 = disable

1 = enable without check digit

char 4 -> code EAN13 conversion

0 = disable

1 = enable without check digit

char 5 -> code EAN8 conversion

0 = disable

1 = enable without check digit

char 6 -> code UPCA conversion

0 = disable

1 = enable without check digit

char 7 -> code UPCE conversion

0 = disable

1 = enable without check digit

char 8 -> code 3/9 Full conversion

0 = disable

1 = enable without check digit

3 = enable with check digit

This function returns TRUE.

int HOcrGetDibPtr (const char *FilePath)

Returns a pointer to the DIB (or NULL in case of error).
FilePath (es: "c:\00001.bmp") is a BMP file. This function reads only 256 gray levels images as all the engines need that type of format to perform recognition.

Example program

The “example.exe” helps to understand each library function.

Important: Please verify the read-only attribute of all files. Remove read-only attribute if enabled.

Important: 1D Barcodes is the only engine that does not need a license file.

OMNIFONT codeline

Use the following instructions to set up the program to use OMNIFONT and to verify the settings are enabled.

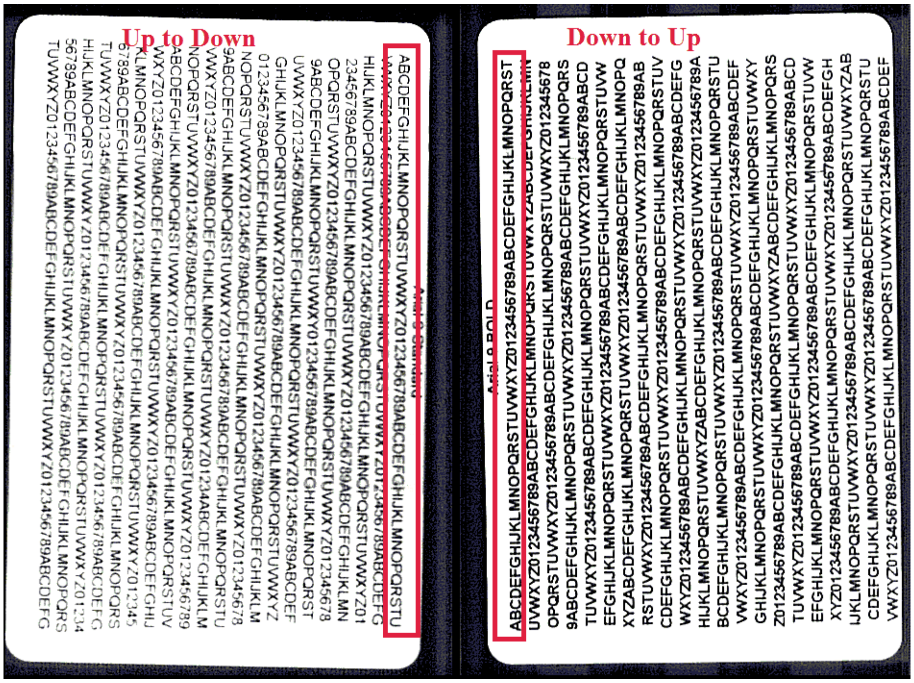
1. Start the “**Example.exe**” program located in the release directory.
    a. Select the recognition Type (**TYPE=OMNIFONT**) .
    b. Load the program with “**LoadRecognizer**” and then click “**I Agree**”.
    c. The “**Loaded**” message will appear.
    d. If you have a demo license a pop up window will appear.
    e. If the demo license grace period is expired or you do not have a valid license file the recognizer will **not** load.
2. Load the “**Example.bmp**” image using the “**ReadImage**” button. “Example.bmp” is located in the Img directory.
3. Select the recognition type (**Type=OMNIFONT).**
4. Click “**MakeOCR”**.
5. A window with the recognized text will appear.

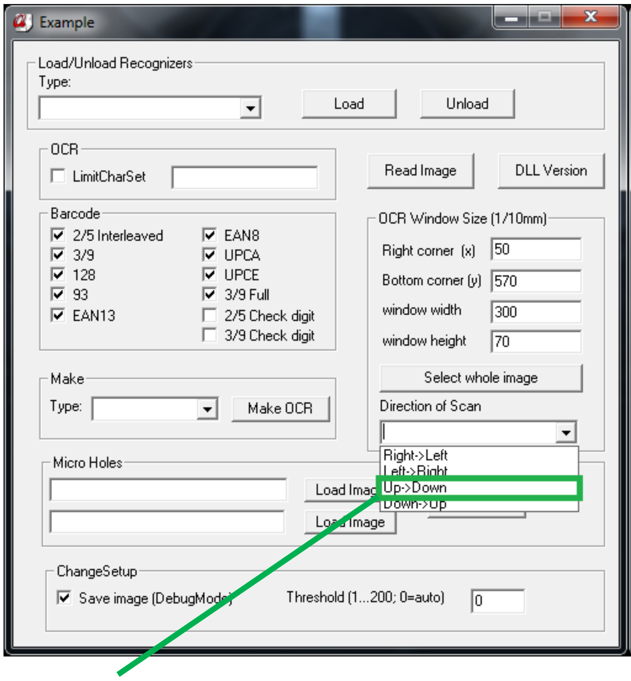
E13B codeline

Use the following instructions to set up the program to use E13B and to verify the settings are enabled.

1. State the “**Example.exe”** program located in the release directory.
    a. Select the recognition type (**type=E13B**).
    b. Load the program with ”**LoadRecognizer”** and then click “I Agree”.
    c. The “**Loaded**” message will appear.
    d. If you have a demo license a pop up window will appear.
    e. If the demo license grace period is expired or you do not have a valid license file the recognizer will **not** load.
2. Load the “**Example.bmp**” image using the “**ReadImage**” button. “Example.bmp” is located in the Img directory.
3. In the “**OCR Window Size**” area, write:
    a. Y=50
    b. X=300
    c. Height=70
    d. Width=1150
4. Select the recognition type (Type=E13B).
5. Click “**MakeOCR**”.
6. Three images will be displayed:
    a. The original
    b. The selected are in the original grayscale
    c. The selected area converted to B/W.
7. A window with the recognized text will appear.

APPENDIX A





Example

Load/Unload Recognizers
Type: [ ]   Load   Unload

OCR
☐ LimitCharSet [ ]

Read Image   DLL Version

Barcode
☑ 2/5 Interleaved ☑ EAN8
☑ 3/9 ☑ UPCA
☑ 128 ☑ UPCE
☑ 93 ☑ 3/9 Full
☑ EAN13 ☐ 2/5 Check digit
☐ 3/9 Check digit

OCR Window Size (1/10mm)
Right corner (x) 50
Bottom corner (y) 570
window width 300
window height 70

Select whole image

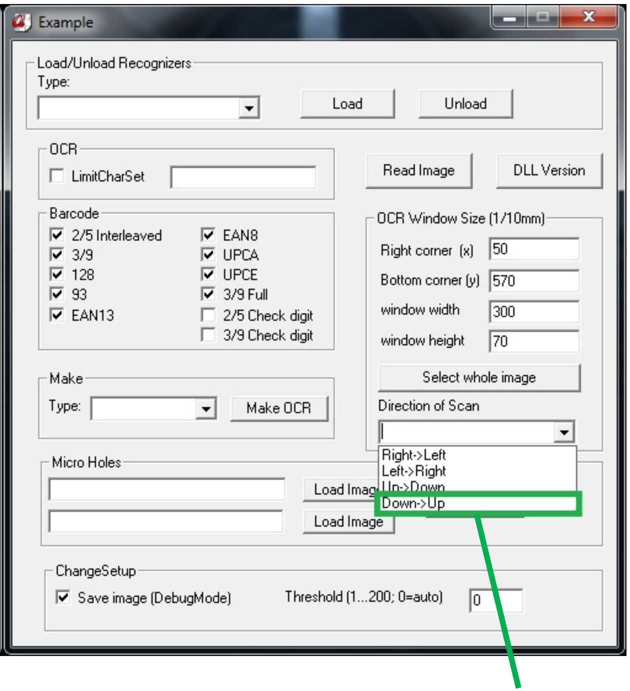
Direction of Scan
Right->Left
Left->Right
Up->Down
Down->Up

Make
Type: [ ]   Make OCR

Micro Holes
Load Image
Load Image

ChangeSetup
☑ Save image (DebugMode)   Threshold (1...200; 0=auto) 0

Up to Down



Example

Load/Unload Recognizers
Type: [ ]   Load   Unload

OCR
☐ LimitCharSet [ ]

Read Image   DLL Version

Barcode
☑ 2/5 Interleaved ☑ EAN8
☑ 3/9 ☑ UPCA
☑ 128 ☑ UPCE
☑ 93 ☑ 3/9 Full
☑ EAN13 ☐ 2/5 Check digit
☐ 3/9 Check digit

OCR Window Size (1/10mm)
Right corner (x) 50
Bottom corner (y) 570
window width 300
window height 70

Select whole image

Direction of Scan
Right->Left
Left->Right
Up->Down
Down->Up

Make
Type: [ ]   Make OCR

Micro Holes
Load Image
Load Image

ChangeSetup
☑ Save image (DebugMode)   Threshold (1...200; 0=auto) 0

Down to Up