

OCRDLL

(ClassOCR.dll)

Standard Version

Version 2.10

User Guide

25 May 2010

For latest version of the user guide please go to www.informatik.com/manuals.html

Table of Contents

Summary Instructions	3
Demo Version, Licensed Version	4
Creating your Project	4
Code Samples:	6
Language Files	12
Technical Support	13
License, Warranty, Disclaimer	13
Copyright	15
Tesseract Copyright and License Notice	15

Summary Instructions

Please make sure that you have the latest version of the program. Please visit www.informatik.com.

The program allows you to build OCR functions into your .NET application.

The program requires Microsoft Framework 3.5 or higher.

The following file **graphics formats** are supported: TIFF, PNG, BMP, GIF, JPEG.

There are two versions: **Standard** and **Basic**. Both versions support TIFF, BMP, PNG and JPEG files. Only the Standard version supports PDF (most formats) and JPEG-compressed TIFF formats. There are differences in licensing terms. This user guide is for the Standard Version.

You can OCR the entire image or a specified rectangle area of the image. Instead of passing a source file name, you can also specify a memory bitmap. If you pass a bitmap, use the keyword 'bitmap' (exact spelling, see code sample below. If you capture a single words or short text strings, the rendition may be faulty, in particular if the word has no ascent characters (only low characters such as m, n, o s, and no high characters like h, t, f, etc.)

For **multi-page files**, you need to call the OCRDLL (ClassOCR.dll) for each page. The page number must be specified as an argument in the function call (1 = page 1). Sample code below illustrates how you can iterate through the pages.

You may want to specify allowed characters. For example, to limit the output to numeric characters, set the **_LimitedCharacterSet** property to `_LimitedCharacterSet="0123456789"`.

The result is returned as a text variable. The layout is basic.

The function returns zero (0) if successful, else it returns an error code. In your code you should always read the return code.

If the output has many errors (100% accuracy is rare), set the **_AccuracyPlus** property to `_AccuracyPlus=True`, as in the main sample below. Accuracy may be improved but the process will be slower.

The OCR function runs for a specified language. Available **languages** are: English, German, French, Italian, Spanish, Portuguese and Dutch. If the language is not specified, the system uses the English language.

Each language requires certain Tessdata **language files**. The language files can be downloaded from the www.informatik.com/ocr.html website. After downloading the language files (zip file) for a specific language, place the unzipped files in a dedicated folder. It is recommended that you make it a subfolder of the application's program folder. A compulsory argument in the function call will refer to the path of that folder (see samples below). If you download the language files for several languages, they all can be in the same folder.

You can add a **progress bar**. See sample code below.

A **sample project** can be downloaded from www.informatik.com/ocr.html. The sample project is simplified (just for English, without barcodes).

Demo Version, Licensed Version

The trial version adds demo lines at the top and bottom of the output text and displays a demo message. The demo version must be used only for testing and must not be used commercially. After 30 days, un-install the program or purchase a license. If you have a licensed version enter the license code in the RegistrationCode argument line; see sample below.

Creating your Project

- In your project you must reference the ClassOCR.dll file in Project -> Add References.
- When compiling your project, set the CPU Target to x86 so that your application also runs on 64-bit machines. The setting is made in Project -> Properties -> Compile -> Advanced Compile Options.
- The directory that runs your application executable must have the following four files:

- ClassOCR.dll
- tessnet2_32.dll
- O2S.Components.PDFRender4NET.dll
- Tifftek32.dll
- Progressbar.exe, if used
- Folder with the necessary language files. See details above.

- When **deploying your application** you must include the following files:

ClassOCR.dll
tessnet2_32.dll
O2S.Components.PDFRender4NET.dll
Tifftek32.dll

Progressbar.exe, if used
Language files in separate folder. See details below.

The files must be placed in the same folder as your application executable.

Also create and deploy the folder with the necessary language files.

- Please review the codes samples below.

All necessary files can be found in C:\Program Files\Informatik Inc\OCR DLL... folder.

A **sample project** can be downloaded from www.informatik.com/ocr.html. The sample project is simplified (just for English, without barcodes).

Code Samples:

OCR an image file

VB.NET

```
'Add this code in your test code to diagnose problems
Dim checkfile As String = Application.StartupPath & "\tessnet2_32.dll"
If IO.File.Exists(checkfile) = False Then
    MessageBox.Show(checkfile & " file not found in directory. Please refer to user guide for
required files.")
    Exit Sub
End If
'-----
```

```
Dim myobj As New ClassOCR.OCR
Dim filename as string = "C:\somedir\somefrile.tif"
myobj._OpenFile.FileName = filename
myobj._OpenFile.PageSelection = 1
'myobj._OpenFile.Password = "" (optional for PDF files)
myobj._OCRLanguage = ClassOCR.OCRLanguage.English
myobj._OCRTessdataDirectory = Application.StartupPath & "\tessdata"
myobj._RegistrationCode = ""
'myobj._AccuracyPlus = True
Dim result As Integer = myobj.OCR_Read()
If result <> 0 Then
    MessageBox.Show("Error: " & myobj.GetError(result))
    End
Else
    Dim ocrtext as string = myobj._OCRText
    '..Add your code here to process the OCR text.
    MessageBox.Show(ocrtext)
End If
myobj = Nothing
```

C-Sharp

'Add this code in your test code to diagnose problems

```
string checkfile = Application.StartupPath + "\\tessnet2_32.dll";
if (IO.File.Exists(checkfile) == false)
{
    MessageBox.Show(checkfile + " file not found in directory. Please refer to user guide for
required files.");
    return;
}
'-----
```

```
ClassOCR.OCR myobj = new ClassOCR.OCR();
string filename = "c:\\myfile.tif";
myobj._OpenFile.Filename = filename;
myobj._OpenFile.PageSelection = 2;
//myobj._OpenFile.Password = "" (optional for PDF files)
myobj._OCRLanguage = ClassOCR.OCRLanguage.English;
myobj._OCRTessdataDirectory = Application.StartupPath + "\\tessdata";
myobj._RegistrationCode = "";
//myobj._AccuracyPlus = True;
int result = myobj.OCR_Read();
if (result != 0)
{
    MessageBox.Show("Error: " + myobj.GetError(result));
}
else
{
    MessageBox.Show(myobj._OCRText);
}
myobj = null;
```

OCR a Bitmap

VB.NET

(Assuming that in your project you have a bitmap called bmpxyz)

```
Dim myobj As New ClassOCR.OCR
myobj.Sourcebitmap = bmpxyz.clone
myobj._OpenFile.Filename = "bitmap"
myobj._OCRLanguage = ClassOCR.OCRLanguage.English
myobj._OCRTessdataDirectory = Application.StartupPath & "\\tessdata"
myobj._RegistrationCode = ""
Dim result As Integer = myobj.OCR_Read()
If result <> 0 Then
    MessageBox.Show("Error: " & myobj.GetError(result))
End
Else
    Dim ocrtext as string = myobj._OCRText
    '..Add your code here to process the OCR text.
    MessageBox.Show(ocrtext)
End If
```

myobj = Nothing

C-Sharp

(Assuming that in your project you have a bitmap called bmpxyz)

```
ClassOCR.OCR myobj = new ClassOCR.OCR();
myobj.Sourcebitmap = bmpxyz.clone;
myobj._OpenFile.FileName = "bitmap";
myobj._OCRLanguage = ClassOCR.OCRLanguage.English;
myobj._OCRTessdataDirectory = Application.StartupPath + "\\tessdata";
myobj._RegistrationCode = "";
int result = myobj.OCR_Read();
if (result != 0)
{
    MessageBox.Show("Error: " + myobj.GetError(result));
}
else
{
    string ocrtext = myobj._OCRText;
    //..Add your code here to process the OCR text.
    MessageBox.Show(ocrtext);
}
myobj = null;
```

OCR a multi-page TIFF or PDF file

VB.NET

```
Dim myobj As New ClassOCR.OCR
Dim filename as string = "C:\somedir\somefrile.tif"
Dim nrpages As Integer = myobj.GetPages(filename)
If nrpages < 0 Then
    MessageBox.Show("Error: " & myobj.GetError(nrpages))
End
End If
Dim fulltext As String = ""
Dim result As Integer
For p As Integer = 1 To nrpages
    myobj = New ClassOCR.OCR
    myobj._OCRLanguage = ClassOCR.OCRLanguage.English
    myobj._OCRTessdataDirectory = Application.StartupPath & "\tessdata"
    myobj._RegistrationCode = ""
    myobj._OpenFile.FileName = filename
    myobj._OpenFile.PageSelection = p
    result = myobj.OCR_Read()
    If result <> 0 Then
        MessageBox.Show("Error: " & myobj.GetError(result))
    End
Else
    If p = 1 Then
        fulltext = fulltext & (myobj._OCRText)
    Else
        fulltext = fulltext & ControlChars.FormFeed & (myobj._OCRText)
    End If
End If
myobj = Nothing
Next
'.. Add code here to process the output text.
MessageBox.Show(fulltext)
```

C-Sharp

```
ClassOCR.OCR myobj = new ClassOCR.OCR();
string filename = "c:\\myfile.tif";
int nrpages = myobj.GetPages(filename);
int result = 0;
if (nrpages < 0)
{
    MessageBox.Show("Error: " + myobj.GetError(nrpages));
}
string fulltext = "";
for (int p = 1; p <= nrpages; p++)
{
    myobj = new ClassOCR.OCR();
    myobj._OCRLanguage = ClassOCR.OCRLanguage.English;
    myobj._OCRTessdataDirectory = Application.StartupPath + "\\tessdata";
    myobj._RegistrationCode = "";
    myobj._OpenFile.Filename = filename;
    myobj._OpenFile.PageSelection = p;

    result = myobj.OCR_Read();
    if (result != 0)
    {
        MessageBox.Show("Error: " + myobj.GetError(result));
        System.Environment.Exit(0);
    }
    else
    {
        if (p == 1)
        {
            fulltext = fulltext + (myobj._OCRText);
        }
        else
        {
            fulltext = fulltext + "\f" + (myobj._OCRText);
        }
    }
    myobj = null;
}
MessageBox.Show(fulltext);
```

OCR with Progress Bar

VB.NET

```
‘..Code as above  
‘..Progressbar code must be inserted before the myobj.OCR_Read().  
‘..The progressbar.exe file must be in the same folder as your application’s executable.
```

```
Dim proc As New System.Diagnostics.Process  
Dim progressbarapp As String = Application.StartupPath & "\\progressbar.exe"  
If IO.File.Exists(progressbarapp) Then  
    proc.StartInfo.FileName = progressbarapp  
    proc.Start()  
Else  
    MessageBox.Show(progressbarapp & " file not found.")  
End If
```

```
Dim result As Integer = myobj.OCR_Read()  
‘..More code here, as above
```

C-Sharp

```
‘..Code as above  
‘..Progressbar code must be inserted before the myobj.OCR_Read().  
‘..The progressbar.exe file must be in the same folder as your application’s executable.
```

```
System.Diagnostics.Process proc = new System.Diagnostics.Process();  
string progressbarapp = Application.StartupPath + "\\progressbar.exe";  
if (System.IO.File.Exists(progressbarapp))  
{  
    proc.StartInfo.FileName = progressbarapp;  
    proc.Start();  
}  
else  
{  
    MessageBox.Show(progressbarapp + " file not found.");  
}  
Dim result As Integer = myobj.OCR_Read()  
‘..More code here, as above
```

OCR a Specified Rectangle

VB.NET

```
'..Code as above  
'..The rectangle can be specified in Inches or in Pixels.  
'..For Centimeters use Inches divided by 2.54.  
'..Rectangle must be inserted before the myobj.OCR_Read().
```

```
myobj._RectanglePixels.X = 100  
myobj._RectanglePixels.Y = 100  
myobj._RectanglePixels.Width = 400  
myobj._RectanglePixels.Height = 200
```

```
'or  
'myobj._RectangleInches.X = 1  
'myobj._RectangleInches.Y = 1.25  
'myobj._RectangleInches.Width = 3.5  
'myobj._RectangleInches.Height = 1
```

```
Dim result As Integer = myobj.OCR_Read()  
'..More code here, as above
```

C-Sharp

```
'..Code as above  
'..The rectangle can be specified in Inches or in Pixels.  
'..For Centimeters use Inches divided by 2.54.  
'..Rectangle must be inserted before the myobj.OCR_Read().
```

```
{  
    myobj._RectanglePixels.X = 100;  
    myobj._RectanglePixels.Y = 100;  
    myobj._RectanglePixels.Width = 400;  
    myobj._RectanglePixels.Height = 200;  
  
    //or  
    //myobj._RectangleInches.X = 1  
    //myobj._RectangleInches.Y = 1.25  
    //myobj._RectangleInches.Width = 3.5  
    //myobj._RectangleInches.Height = 1  
  
    int result = myobj.OCR_Read();  
}  
//..More code here, as above
```

Language Files

The OCR function runs for a specified language. Available languages are: English, German, French, Italian, Spanish, Portuguese and Dutch. If the language is not specified, the system uses the English language.

Each language requires certain Tesseract language files. The language files can be downloaded from the www.informatik.com/ocr.html website. After downloading the language files (zip file) for a specific language, place the unzipped files in a dedicated folder. It is recommended that you make it a subfolder of the application's program folder. A compulsory argument in the function call will refer to the path of that folder (see samples above). If you download the language files for several languages, they all can be in the same folder.

Technical Support

Please see the contact information shown in the www.informatik.com web site. Support can only be given for program interface and code issues. The program uses Tesseract OCR engine; please understand that and no technical support for it can be given (like misreading of text, etc).

A **sample project** can be downloaded from www.informatik.com/ocr.html. The sample project is simplified (just for English, without barcodes).

License, Warranty, Disclaimer

Please read the terms carefully before installing and using the software, as such conduct will indicate your acceptance of all of the terms of this license agreement. If you do not agree with the terms, the software cannot be licensed to you and you must un-install and return the software to Informatik Inc, or its supplier or distributor.

This License Agreement is a legal agreement between Informatik Inc. ("Licensor"), a Delaware Corporation, and you, the user ("Licensee"), and is effective the date Licensee installs the software.

This Agreement covers all materials associated with the OCR software (ClassOCR.dll), including, without limitation, the downloadable software product, online documentation, and any additional supporting electronic files (herein, the "Software").

The evaluation version may be used for 30 days after installation. It is unlawful to use the software after the 30 day evaluation period without licensing the software and paying the license fees. If a license is not obtained before the expiration of the 30 day evaluation period, the Software must be un-installed and destroyed.

1. GRANT OF LICENSE

Licensor hereby grants to you, and you accept, a nonexclusive license to use the Software according to the following condition:

You may use the Software on one (1) developer's computer (PC or workstation, excluding servers) owned, leased, or otherwise controlled by you for personal or business purposes, and only as authorized in this License Agreement. The Software may not be used on other computers, nor may it be used by, or transferred to, other computers over a network.

A license is required for each developer. The compiled applications developed by the licensed developer, excluding web applications, together with the required support files can be freely distributed as follows: a) the Standard Version covers all users of the developers' company/enterprise (no external sales of the applications); b) for the Basic Version: all users, including external sales of the applications. The Software must be a minor part of your application and your application must not be a programming tool or a wrapper program. Your application must not be a web application. You may not distribute this help file.

2. LICENSOR'S RIGHTS

Licensee acknowledges and agrees that the Software is proprietary to Licensor and protected under international copyright law. Licensee further acknowledges and agrees that all right, title, and interests in and to the Software, including associated intellectual property rights, are and shall remain with Licensor. The License Agreement does not convey to Licensee an interest in or to the Software, but only a limited right of use that may be revoked in accordance with the terms of this License Agreement.

3. OTHER RESTRICTIONS

This License Agreement strictly forbids distribution of the Software with Licensee's application. Distribution of the Software with Licensee's application requires separate authorization and the payment of license fees.

Licensee agrees to make no more than one (1) back-up copy of the Software. Licensee agrees not to assign, sublicense, transfer, pledge, lease, rent, or share the rights assigned under this License Agreement. Licensee agrees not to reverse assemble, reverse compile, or otherwise translate the Software.

4. TERM

This License Agreement is effective when Licensee installs the Software and shall terminate only if the terms of this License Agreement are broken. Licensee agrees to destroy the Software upon termination of this License Agreement.

5. NO WARRANTY; LIMITATION OF LIABILITY

LICENSEE ACKNOWLEDGES THAT THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTY OF ANY KIND. LICENSOR MAKES NO

REPRESENTATIONS OR WARRANTIES REGARDING THE USE OR PERFORMANCE OF THE SOFTWARE. LICENSOR (incl. DEVELOPER, COPYRIGHTHOLDER, DISTRIBUTOR) EXPRESSLY DISCLAIMS THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. LICENSOR SHALL HAVE NO LIABILITY TO LICENSEE OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE CAUSED, DIRECTLY OR INDIRECTLY, BY THE SOFTWARE, INCLUDING, BUT NOT LIMITED TO, ANY INTERRUPTION OF SERVICES, LOSS OF BUSINESS, LOSS OF DATA OR SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES.

6. GOVERNING LAW

This License Agreement shall be construed and governed in accordance with the laws of Pennsylvania.

7. SEVERABILITY

Should any court of competent jurisdiction declare any term of this License Agreement void or unenforceable, such declaration will have no effect on the remaining terms hereof.

8. NO WAIVER

The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

Copyright

Copyright 2009 Informatik Inc. All Rights Reserved.

Tesseract Copyright and License Notice

The program uses the TESSERACT (unaltered) free OCR engine distributed under the Apache V2.0 license.

TESSERACT Copyright and License Notice:

Copyright Protected and Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Informatik OCR Copyright Notice:
Graphics programming, text handling and user interface (excluding Tesseract).
Copyright 2009-2010 Informatik Inc. All Rights Reserved. Supplied AS IS
without any liability