
Aquaforest OCR SDK for .NET Reference Guide



Version 2.20
April 2017

Contents

1	INTRODUCTION.....	1
1.1	SDK OVERVIEW	1
1.1.1	Licensing	1
1.1.2	Folders	2
1.2	TECHNICAL SUPPORT	2
2	AQUAForest OCR MODULE	3
2.1	OVERVIEW	3
2.2	SYSTEM REQUIREMENTS	3
2.2.1	Supported Environments.....	3
2.2.2	.NET Framework.....	3
2.2.3	Visual C++ Runtime.....	3
2.3	APPLICATION DEVELOPMENT AND DEPLOYMENT.....	3
2.3.1	Prerequisites.....	3
2.3.2	References.....	3
2.3.3	Deploying C# and VB.NET Applications	4
2.3.4	Deploying ASP.NET Applications.....	4
2.3.5	Licensing	4
2.4	SAMPLE APPLICATIONS	4
2.5	A SIMPLE EXAMPLE	4
2.5.1	References.....	4
2.5.2	Classes.....	4
2.5.3	Processing Step	5
2.5.4	C#Example	6
2.5.5	VB.NET Example.....	7
3	AQUAForest OCR MODULE API REFERENCE.....	8
3.1	PREPROCESSOR CLASS	8
3.1.1	Constructor.....	8
3.1.2	Properties	8
3.2	OCR CLASS.....	9
3.2.1	Constructor.....	9
3.2.2	Properties and Methods.....	9
3.2.3	Events	9
3.2.4	Subscribing to StatusUpdate using C#.....	9
3.2.5	Subscribing to StatusUpdate using VB.NET	10
3.3	STATUSUPDATEEVENTARGS CLASS	10
3.3.1	Constructor.....	10
3.3.2	Properties	10
3.3.3	Words Class.....	11
3.3.4	Constructor.....	11
3.3.5	Properties	11
3.3.6	Methods	11
3.3.7	WordData Class	11
3.3.8	Properties	11
3.4	ERROR HANDLING.....	12
3.5	PDFMERGER CLASS	12
3.5.1	Constructor.....	12

3.5.2	Methods.....	12
3.6	DISPOSAL AND TEMPORARY FILES FOLDERS	12
3.7	MULTI-THREADED APPLICATIONS	12
3.8	ADVANCED PRE-PROCESSING.....	12
3.9	PROPERTIES FILE.....	15
4	AQUAForest EXTENDED OCR MODULE	18
4.1	OVERVIEW.....	18
4.2	SYSTEM REQUIREMENTS	18
4.2.1	Supported Environments.....	18
4.2.2	.NET Framework.....	18
4.2.3	Visual C++ Runtime.....	18
4.3	FOLDERS.....	18
4.4	APPLICATION DEVELOPMENT AND DEPLOYMENT.....	19
4.4.1	Prerequisites.....	19
4.4.2	References.....	19
4.4.3	Properties.xml	19
4.4.4	Deploying C# and VB.NET Applications	20
5	AQUAForest EXTENDED OCR MODULE API REFERENCE	21
5.1	PREPROCESSOR CLASS	21
5.1.1	Constructor.....	21
5.1.2	Properties.....	21
5.2	OCR CLASS.....	21
5.2.1	Constructor.....	21
5.2.2	Properties and Methods.....	21
5.2.3	Events	21
5.2.4	Subscribing to StatusUpdate	21
5.3	STATUSUPDATEEVENTARGS CLASS	22
5.3.1	Constructor.....	22
5.3.2	Properties	22
5.3.3	Words Class.....	22
5.3.4	Words Constructor	23
5.3.5	Words Properties	23
5.3.6	Words Methods	23
5.3.7	WordData Class	23
5.3.8	WordData Properties.....	23
5.3.9	CharacterData Class	23
5.3.10	CharacterData Properties	24
5.4	SUPPORTED LANGUAGES	24
6	AQUAForest OCR VS. EXTENDED OCR	29
6.1	DIFFERENCES BETWEEN AQUAForest OCR AND EXTENDED OCR	29
6.1.1	References.....	29
6.1.2	Ocr Methods	29
6.1.3	Ocr Properties.....	29
6.1.4	PreProcessor Properties	30
6.2	CREATING A SIMPLE APPLICATION	31
6.2.1	Using Aquaforest SDK and Visual Studio 2012+	32
6.2.2	Converting to Extended OCR	42
7	BARCODE MODULE	47
7.1	OVERVIEW.....	47

7.1.1	Supported Barcode Formats.....	47
7.2	SYSTEM REQUIREMENTS	47
7.2.1	Supported Environments.....	47
7.2.2	.NET Framework.....	47
7.3	APPLICATION DEVELOPMENT AND DEPLOYMENT.....	47
7.3.1	Prerequisites.....	47
7.3.2	References.....	48
7.3.3	Deploying C# and VB.NET Applications	48
7.3.4	A Simple C# Example.....	48
7.3.5	Barcode Module API	49
8	BACKGROUND - SEARCHABLE PDFS	50
8.1	WHAT IS A SEARCHABLE PDF?	50
8.2	INSIDE A SEARCHABLE PDF	50
8.3	OCR ACCURACY	50
8.3.1	Original Image Quality.....	50
8.3.2	Image DPI and Format.....	50
8.3.3	Despeckle	51
8.3.4	Deskew.....	51
8.3.5	Auto-Rotate.....	51
8.3.6	Graphics Areas.....	51
8.3.7	Language Settings.....	51
8.4	HARDWARE AND PERFORMANCE	51
8.4.1	CPU Power	51
8.4.2	Exploiting Multiple CPUs	51
8.4.3	Memory.....	51
9	ACKNOWLEDGEMENTS	52

1 Introduction

1.1 SDK Overview

The Aquaforest OCR SDK for .NET incorporates the same high performance OCR engine that is included in our Aquaforest TIFF Junction, Autobahn DX and Aquaforest Searchlight products.

The SDK API allows developers full control over OCR processing to enable customized integration of OCR within .NET applications.

- OCR PDF, TIFF, BMP, PNG or JPEG files.
- Create Searchable PDF, RTF, DOCX, HTML, CSV or Text output files.
- Control pre-processing options such as despeckle, deskew, line removal and autorotate.
- Select from over 100 supported document languages.
- Enumerate the OCR results, examining the words and characters recognized along with their coordinates.
- Blank page removal.
- Process multi-page TIFF and PDF files one page at a time or all in one operation.
- Decode barcodes.
- Split documents by barcode.
- Perform parallel processing using multi-threading.
- Multiple PDF version support.
- Support for multiple language within a single document from the same character set.
- Intelligent High Quality Compression.

1.1.1 Licensing

There are a few changes in the way this release is licensed; this is to offer buyers a higher flexibility. The table below shows a breakdown of the licensing.

Function	Basic Edition	Standard Edition	Advanced Edition	Extended Edition
OCR from Bitmap or TIFF	✓	✓	✓	✓
Image Pre-Processing and Auto-Rotation	✓	✓	✓	✓
Support for 23 Languages	✓	✓	✓	✓
.NET Programmatic / Zonal Access to results	✓	✓	✓	✓
Txt / RTF Output	✓	✓	✓	✓
1 Thread	✓	✓	✓	✓
Blank Page Removal	✓	✓	✓	✓
PDF Merging	✓	✓	✓	✓
Barcode Decoding	✓	✓	✓	✓
PDF Input		✓	✓	✓
Searchable PDF Output		✓	✓	✓
2 Threads		✓	✓	✓

Function	Basic Edition	Standard Edition	Advanced Edition	Extended Edition
Unlimited Threads			✓	✓
Advanced MRC Compressed PDF Output			✓	✓
Advanced Pre-Processing			✓	✓
Support for 129 languages				✓
Support for multiple languages within a single document from the same character set				✓
Asian language support (<i>Optional add-on</i>)				✓
Arabic language support (<i>Optional add-on</i>)				✓
Intelligent High Quality Compression (<i>Optional add-on</i>)				✓
Multiple document output formats: CSV, DOCX, EPUB, EXCELML, HTML, OPEN DOCUMENT TEXT, PDF, RTF, TXT, WORDML, XLSX and XPS				✓
Multiple PDF version output support				✓

1.1.2 Folders

The SDK contains the following folders:

- *bin* – This contains all the assemblies, DLLs and configurations files for the Standard OCR
- *docs* – contains the documentation of the SDK
- *license* – Licensing information
- *redistributables* – C++ redistributables, Aquaforest OCR SDK Prerequisite Check.exe
- *samples* – Standard OCR samples in C#, VB.NET and ASP.NET
- *xbin* – This contains all the assemblies, DLLs and configuration files for the Extended OCR
- *xsamples* – Extended OCR samples

1.2 Technical Support

Please contact Aquaforest Technical Support with any queries by email at support@aquaforest.com. If required, telephone support is also available; please contact Aquaforest using the telephone contact details provided on the company website contact page.

2 Aquaforest OCR Module

2.1 Overview

2.2 System Requirements

2.2.1 Supported Environments

- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2016

2.2.2 .NET Framework

.NET Framework 4.5.2

2.2.3 Visual C++ Runtime

The Visual C++ 2013 Redistributable package is required for deployment as well as development.

2.3 Application Development and Deployment

2.3.1 Prerequisites

The table below shows the prerequisites needed for building applications using the Aquaforest OCR engine.

Application Platform	VC++ Redistributable	Minimum .NET Framework Version	Minimum Visual Studio Version
x86	VC ++ 2013 x86	.NET Framework 4.5.2	Visual Studio 2012
x64	VC ++ 2013 x86 and VC++ 2013 x64		
Any CPU	VC ++ 2013 x86 and VC++ 2013 x64		

There is a diagnostic tool found at "[SDK installation path]\diagnostics\Aquaforest OCR SDK Prerequisite Check.exe" which can be used to check if the correct versions of .NET Framework and Visual C++ Redistributables are installed.

Note: If you are using Visual Studio 2012 or 2013 to build your application(s), you need to ensure that you also have the [.NET Framework 4.5.2 Multi-targeting pack](#) installed so as to be able to build applications that target .NET Framework 4.5.2.

2.3.2 References

To use the API a reference to Aquaforest.Ocr.Api must be included in your application. If you wish to enumerate the OCR results rather than simply generate PDF, RTF or TXT outputs then you will also need to add a reference to Aquaforest.Ocr.Definitions.

2.3.3 Deploying C# and VB.NET Applications

Any deployment method should ensure that the target system meets the requirements (see [section 2.2](#)) and install the Visual C++ 2013 Redistributable package and .NET Version 4.5.2 framework if necessary in addition to the full contents of the SDK **bin** folder.

For building and deploying C# and VB.NET applications, the recommended approach is to specify the full path of the SDK bin folder to the OCR resource folder (`_ocr.ResourceFolder`) as shown in the sample code in sections [2.5.4](#) and [2.5.5](#).

The SDK also contains an in-built functionality to detect if all the required assemblies and files required by the SDK are present. If they are not, an exception will be thrown listing all the files that are missing.

2.3.4 Deploying ASP.NET Applications

The same two approaches that work for C# and VB.NET can also be employed for ASP.NET applications. Note that with trial licenses a pop-up dialog box appears on the server.

2.3.5 Licensing

Production system deployment requires that a license string is defined in the code. The license string defines the number of concurrent OCR processes that can be run.

For example:

```
ocr.License = "MT0xMjM0NTY7BLk4uT3RoZXOzM9NDs0PVRydWEYzMDRFOEQxMzg0QkQ5ODREQtK3RQ";
```

If the string is not specified, the SDK will run in evaluation mode. In evaluation mode:

- A trial "pop-up" will appear for each document processed
- Generated searchable PDFs will include indelible watermarks
- Only 3 pages are generated for text or RTF files.

2.4 Sample Applications

The samples folder includes a number of sample applications in C#, VB.NET and ASP.NET. The solutions provided are all created using Visual Studio 2012 and conversion to Visual Studio 2012 and above is handled automatically by that IDE.

Description of the sample applications are described in the Cookbook found in the "docs" folder.

2.5 A simple example

The full API reference is in [section 3](#) of this guide, but as a starting point a simple example of a C# and VB.NET console application that creates a searchable PDF from a source TIFF file is described in sections [2.5.4](#) and [2.5.5](#).

2.5.1 References

A reference to the Aquaforest.OCR.Api DLL should be added in your application.

If you wish to access the results of the OCR on a word by word basis, for example to obtain word and character results including positional information then you will also need to reference Aquaforest.OCR.Definitions DLL.

2.5.2 Classes

There are two classes used for the OCR:

- **PreProcessor** – This class configures and performs image pre-processing (such as de-skewing images) to ensure optimal OCR performance.
- **OCR** – This is the class that configures and performs the Optical Character Recognition.

Additionally, for accessing the OCR results at an individual word level the following classes are used:

- **Words** – This class contains a collection of words in which is contained all the data available for the words and characters for any given page.
- **WordData** – This class contains a collection of characters that make up the word along with the positional information for each character and the whole word.
- **StatusUpdateEventArgs** – This class is available for each page processed when subscribing to the StatusUpdate event and provides information relating to the processing outcome for the page.

2.5.3 Processing Step

The following steps are involved in this example

1. Create the OCR and PreProcessor objects
2. Specify the location of the OCR bin folder
3. Specify Pre-Processor options
4. Specify OCR Options
5. Read the source file
6. Perform the recognition
7. Save the searchable PDF
8. Delete temporary files (these are by default stored in %TEMP% but the location can be specified using `ocr.TempFolder`)

2.5.4 C# Example

```
using System;
using System.IO;
using Aquaforest.OCR.Api;

namespace ConvertTIFFToSearchablePDF
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                using (Ocr ocr = new Ocr())
                {
                    string resourceFolder = Path.GetFullPath(@"..\..\..\..\bin\");

                    string currentEnvironmentVariables =
                        Environment.GetEnvironmentVariable("PATH");

                    if (!currentEnvironmentVariables.Contains(resourceFolder))
                    {
                        Environment.SetEnvironmentVariable(
                            "PATH", currentEnvironmentVariables + ";" + resourceFolder);
                    }

                    // Set OCR options
                    ocr.ResourceFolder = resourceFolder;
                    ocr.Language = SupportedLanguages.English;
                    ocr.EnableConsoleOutput = true;
                    ocr.EnablePdfOutput = true;

                    // Set PreProcessor options
                    PreProcessor preProcessor = new PreProcessor();
                    preProcessor.Deskew = true;
                    preProcessor.Autorotate = false;

                    // Read source TIFF file
                    ocr.ReadTIFFSource(
                        Path.GetFullPath(@"..\..\..\..\documents\source\sample.tif"));

                    // Perform OCR recognition
                    if (ocr.Recognize(preProcessor))
                    {
                        // Save output as searchable PDF
                        ocr.SavePDFOutput(
                            Path.GetFullPath(@"..\..\..\..\documents\output\sample.pdf"), true);
                    }

                    ocr.DeleteTemporaryFiles();
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("Error in OCR Processing : " + e.Message);
            }
        }
    }
}
```

2.5.5 VB.NET Example

```
Imports System.IO
Imports Aquaforest.OCR.Api

Module Module1

    Sub Main()
        Try

            Using ocr As New Ocr()

                Dim resourceFolder As String = Path.GetFullPath("../..\\..\\..\\..\\bin")

                If Not Environment.GetEnvironmentVariable("PATH").Contains(resourceFolder) Then
                    Environment.SetEnvironmentVariable(
                        "PATH", Environment.GetEnvironmentVariable("PATH") + ";" + resourceFolder)
                End If

                'Set OCR options
                ocr.ResourceFolder = resourceFolder
                ocr.EnableConsoleOutput = True
                ocr.Language = SupportedLanguages.English
                ocr.EnablePdfOutput = True

                ' Set PreProcessor options
                Dim preProcessor As New PreProcessor()
                preProcessor.Deskew = True
                preProcessor.Autorotate = False

                'Read source TIFF file
                ocr.ReadTIFFSource(Path.GetFullPath("../..\\..\\..\\documents\\source\\sample.tif"))

                'Perform OCR recognition
                If ocr.Recognize(preProcessor) Then

                    'Save output as searchable PDF
                    ocr.SavePDFOutput(
                        Path.GetFullPath("../..\\..\\..\\documents\\output\\sample.pdf"), True)

                End If

                ocr.DeleteTemporaryFiles()

            End Using

            Catch ex As Exception
                Console.WriteLine("Error in OCR Processing :" + ex.Message)
            End Try

        End Sub

    End Module
```

3 Aquaforest OCR Module API Reference

To use the API a reference to Aquaforest.Ocr.Api must be included in your application. If you wish to enumerate the OCR results rather than simply generate PDF, RTF or TXT outputs then you will also need to add a reference to Aquaforest.Ocr.Definitions.

3.1 PreProcessor Class

A PreProcessor object, which must be created and passed to the `Ocr` object, controls all of the pre-processing that can be performed on the input image in order to improve the quality of the output. Instantiation of the PreProcessor object will initialise a default set of pre-processing options which result in minimal image manipulation. For a full description of the pre-processing options available and appropriate values see [section 3.1.2](#) Properties below.

3.1.1 Constructor

```
PreProcessor preProcessor = new PreProcessor();
```

3.1.2 Properties

Refer to the **OCRSDK 2.2 (Aquaforest engine).chm** file found in the folder “[SDK installation path]\docs\help” to view all the properties available in the PreProcessor class.

3.2 OCR Class

The OCR object is used to control OCR processing, obtain status updates during processing and retrieve the resulting output from this processing upon completion.

3.2.1 Constructor

```
Ocr ocr = new Ocr();
```

3.2.2 Properties and Methods

Refer to the **OCRSDK 2.2 (Aquaforest engine)** file found in the folder “[SDK installation path]\docs\help” to view all the properties and methods available in the Ocr class.

3.2.3 Events

Event	Description
void StatusUpdate (object sender, StatusUpdateEventArgs statusUpdateEventArgs)	This event is raised when processing of a page is complete. The StatusUpdateEventArgs object provides access to information relating to the status of the page processed.

3.2.4 Subscribing to StatusUpdate using C#

Include a reference to Aquaforest.OCR.Definitions.dll in the solution and define a method to match the event signature, see below.

```
private void OcrStatusUpdate(object sender, StatusUpdateEventArgs statusUpdateEventArgs)
{
    double confidenceScore = statusUpdateEventArgs.ConfidenceScore;
    // anything confidenceScore below 1 might be worth investigation
    int pageNumber = statusUpdateEventArgs.PageNumber;
    int rotation = statusUpdateEventArgs.Rotation;
    // rotation used in 90° steps from beginning
    // orientation (0), i.e. 1 = 90, 2 = 180, 3 = 270
    bool textAvailable = statusUpdateEventArgs.TextAvailable;
    bool imageAvailable = statusUpdateEventArgs.ImageAvailable;
    bool blankPage = statusUpdateEventArgs.BlankPage;
}
```

Finally add a new reference to the event on the OCR object:
_ocr.StatusUpdate += OcrStatusUpdate;

3.2.5 Subscribing to StatusUpdate using VB.NET

Include a reference to Aquaforest.OCR.Definitions.dll in the solution and define a method to match the event signature, see below.

```
Private Sub OcrPageCompleted(ByVal sender As Object, ByVal statusUpdateEventArgs As
StatusUpdateEventArgs) Handles _ocr.StatusUpdate

    Dim confidenceScore As Double
    Dim pageNumber As Integer
    Dim rotation As Integer
    Dim textAvailable As Integer
    Dim imageAvailable As Integer
    Dim blankPage As Boolean

    confidenceScore = statusUpdateEventArgs.ConfidenceScore
    ' anything confidenceScore below 1 might be worth investigation
    pageNumber = statusUpdateEventArgs.PageNumber
    rotation = statusUpdateEventArgs.Rotation
    ' rotation used in 90° steps from beginning orientation (0), i.e. 1 = 90, 2 = 180, 3 = 270
    textAvailable = statusUpdateEventArgs.TextAvailable
    imageAvailable = statusUpdateEventArgs.ImageAvailable
    blankPage = statusUpdateEventArgs.BlankPage

End Sub
```

Declare the OCR object using "WithEvents":

```
Private WithEvents _ocr As New Ocr
```

3.3 StatusUpdateEventArgs Class

This class contains information relating to the conversion status of a page.

3.3.1 Constructor

An instance of this class is obtained for each page processed when subscribing to the event StatusUpdate.

3.3.2 Properties

Property	Description
int PageNumber	This property returns page for which the object relates to.
int Rotation	A value from 0 to 3 which indicates the rotation used for the output in terms of the number of 90° steps away from the orientation in which the input page was provided. If AutoRotate is set to false this will always be 0.
double ConfidenceScore	Generally a value of 1 or greater would indicate that reasonable OCR of a page, but this should be confirmed using "typical" source files.
bool TextAvailable	This property indicates whether text was extracted for the page.
bool ImageAvailable	This property indicates whether an image (after all appropriate pre-processing) was successfully extracted.
bool BlankPage	This property indicates whether the page was detected as blank.

3.3.3 Words Class

This class contains a collection of WordData objects which are available on a page by page basis.

3.3.4 Constructor

An instance of this class is obtained by calling the ReadPageWords method on the Ocr object, passing the page for which the words are required.

3.3.5 Properties

Property	Description
int Count	This property returns the number of WordData objects in the collection.
int Height	This property returns the height of the current word.
int Width	This property returns the width of the current word.

3.3.6 Methods

Method	Description
WordData GetFirst()	Returns the first WordData object in the collection and sets the index to this item.
WordData GetNext()	Returns the next WordData object in the collection and sets the index to this item.
int GetHeight(int index)	Returns the word height from the WordData object stored at the specified index in the collection.
int GetWidth(int index)	Returns the word width from the WordData object stored at the specified index in the collection.

3.3.7 WordData Class

This class contains the individual characters along with the positional information relating to each character in the word and to the word as a whole.

3.3.8 Properties

Property	Description
float AverageCharacterHeight	This property returns the average height of all the characters in the word.
float AverageCharacterWidth	This property returns the average width of all the characters in the word.
int Bottom	This property returns the bottom of the word.
int CharacterList	This property returns a list of CharacterData objects for the word.
int Height	This property returns the height of the word.
int Left	This property returns the left edge of the word.
int Top	This property returns the Top of the word.
int Width	This property returns the width of the word.
string Word	This property returns the word as a string.

3.4 Error Handling

There are two options regarding error handling using the API.

1. Using the default settings various exceptions can be thrown by the `Ocr` object so these should be trapped within the calling code.
2. Alternatively `HandleExceptionsInternally` can be set to true with the result that method calls will return false on error but throw no exceptions. The calling code can obtain the last exception from the `LastException` property if details of the failure are required.

3.5 PdfMerger Class

This class can be used to merge two PDFs.

3.5.1 Constructor

```
PdfMerger pdfMerger = new PdfMerger("C:\\out\\Merged.pdf");
```

3.5.2 Methods

Method	Description
<code>void Append(string pdfFileToAdd)</code>	Appends the document specified to the in memory PDF document.
<code>void Close()</code>	Writes the output to the file specified in the constructor.
<code>void Dispose()</code>	Clears any resources not yet released. This is useful if <code>Close</code> (which will automatically free such resources) is not called, for example if as a result of an error you do not wish to write the merged output.

3.6 Disposal and Temporary Files folders

During the OCR processing various temporary files are generated and used at different stages. These temporary files can be removed by calling `DeleteTemporaryFiles`. However, such a call should not be made until all processing (both within the `Ocr` object and calling code) on a file is complete as these files are required when calling `SaveRTFOutput`, `SavePDFOutput`, `SaveTextOutput`, `GetPageImage` and `ReadPageWords`. When the `Ocr` object is disposed, the temporary files are automatically removed.

3.7 Multi-threaded applications

Temporary files created and used throughout the OCR processing are named according to the page number, therefore if `Ocr` objects are instantiated in multiple threads then a different temporary folder must be set for each folder. If this is not done then un-expected behaviour will result.

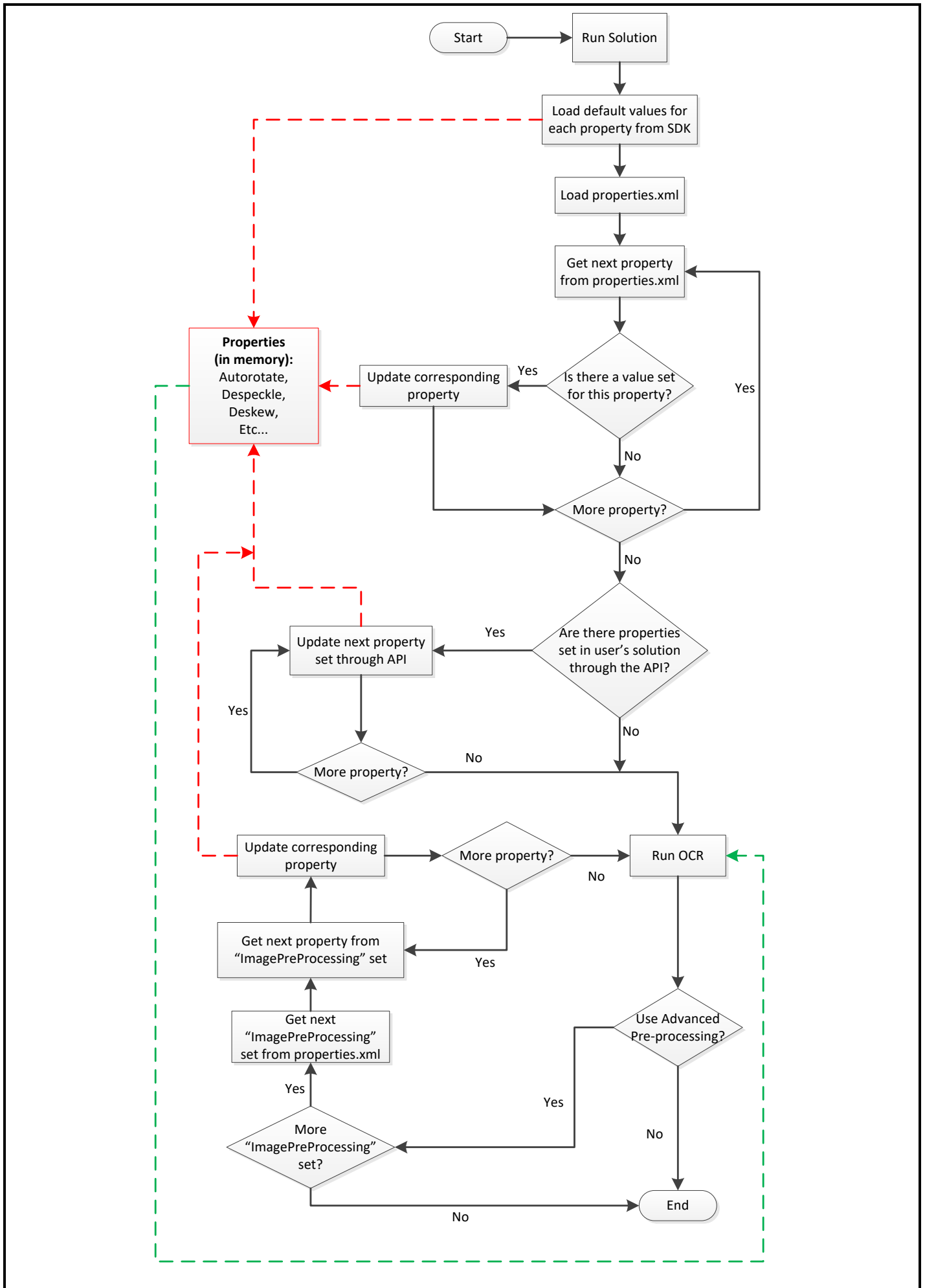
3.8 Advanced pre-processing

When the `AdvancedPreProcessing` property on the OCR object is set to false the OCR and image processing engines will use the settings in the `ImagePreProcessingDefaults` section of the file `Properties.xml` modified by any properties set on the OCR and `PreProcessing` objects.

Setting `AdvancedPreProcessing` to true will enable the use of these default settings first (without modification by the properties set on the OCR and `PreProcessing` objects) followed by the same defaults modified by the values in the `ImagePreProcessing` sections from `ID="1"` to `ID="n"` where `n` is the last consecutive set defined in `Properties.xml`.

Using heuristics and dictionary lookup the quality of the OCR output is then compared in order to determine the optimum set to output. In this way it is possible to define different sets of OCR and pre-processing

conditions that are suited to different types of source documents. This approach can also improve the handling of documents that contain different types of pages, e.g. scanned at different qualities, containing different languages, containing standard and dot matrix prints, etc.



3.9 Properties File

The following are descriptions of those properties in the file Properties.xml that are most likely to be changed to improve engine performance. If you require further information regarding any properties in the file then please contact Aquaforest via support@aquaforest.com for assistance.

Binarize – This setting determines how the image will be converted into a bitonal one for OCR. The following are valid options:

-1 – This utilizes a technique whereby those parts of the image that have certain characteristics indicative of characters are extracted from the underlying image. This approach can give the best results on pages such as magazine images, news print, etc. and will handle light text on darker backgrounds. This approach can cause an increase in processing time with certain images.

0 – This utilizes the binarization capabilities built into the OCR engine and whilst it can give good results in limited situations it is not generally recommended.

>0 – A value greater than 0 (the recommended default is 200) will use a simple threshold technique comparing the intensity of the pixel to the threshold value to determine whether it should be set to black or white. This simple approach is the fastest option.

BoxSize – Setting a value above 0 will cause the removal of enclosing boxes from the image used for the OCR processing. The default recommended is 100, i.e. where the box edges are 100 pixels or greater.

BackgroundFactor - Sampling size for the background portion of the image. The higher the number, the larger the size of the image blocks used for averaging which will result in a reduction in size but also quality. Default value is 3

DotMatrix - Set this to True to improve recognition of dot-matrix fonts. Default value is False. If set to true for non dot-matrix fonts then the recognition can be poor

ForegroundFactor - Sampling size for the foreground portion of the image. The higher the number, the larger the size of the image blocks used for averaging which will result in a reduction in size but also quality. Default value is 3

Jbig2EncFlags – These are the flags that will be passed to the application used to generate JBIG2 versions of images used in PDF generation (assuming this compression is enabled). Options are as follows:

- b <basename>: output file root name when using symbol coding
- d --duplicate-line-removal: use TPGD in generic region coder
- p --pdf: produce PDF ready data
- s --symbol-mode: use text region, not generic coder
- t <threshold>: set classification threshold for symbol coder (def: 0.85)
- T <bw threshold>: set 1 bpp threshold (def: 188)
- r --refine: use refinement (requires -s: lossless)
- O <outfile>: dump thresholded image as PNG
- 2: upsample 2x before thresholding
- 4: upsample 4x before thresholding
- S: remove images from mixed input and save separately
- j --jpeg-output: write images from mixed input as JPEG
- v: be verbose

Language – The acceptable values are as follows:

- 0 - English
- 1 - German
- 2 - French
- 3 - Russian
- 4 - Swedish
- 5 - Spanish
- 6 - Italian
- 7 - Russian English
- 8 - Ukrainian
- 9 - Serbian
- 10 - Croatian
- 11 - Polish
- 12 - Danish
- 13 - Portuguese
- 14 - Dutch
- 19 - Czech
- 20 - Roman
- 21 - Hungarian
- 22 - Bulgarian
- 23 - Slovenian
- 24 - Latvian
- 25 - Lithuanian
- 26 - Estonian
- 27 – Turkish

MaxDeskew – Maximum angle by which a page will be deskewed

Morph – Morphological options that will be applied to the binarized image before OCR. If left blank none is applied. Common options include those listed below but for more options please contact support@aquaforest.com:

- d2.2 – 2x2 dilation applied to all black pixel areas, useful for faint prints.
- e2.2 – 2x2 erosion applied to all black pixel areas, useful for heavy prints.
- c2.2 – closing process that performs a 2x2 dilation followed by a 2x2 erosion with the result that holes and gaps in the characters are filled.

NoPictures - By default, if an area of the document is identified as a graphic area then no OCR processing is run on that area. However, certain documents may include areas or boxes that are identified as “graphic” or “picture” areas but that actually do contain useful text. Setting NoPictures to True will cause it to ignore areas identified as pictures whilst setting it to False will force OCR of areas identified as pictures.

OneColumn - The default value for this is true which improves the handling of single column text. Better handling of multi-column text such as magazine or news print can be achieved.

PdfToImage – The SDK ships with two engines for the conversion of PDF pages to images for OCR. The default engine is used when this is set to 0 but if certain PDF source documents are proving problematic then the alternate engine can be used by changing this value to 1.

PdfToImageIncludeText – When set to False this will prevent the conversion of real text (i.e. electronically generated as opposed to text that is part of a scanned image) from being rendered in the page images extracted from the PDF. This is because the text is already searchable and so generally does not require OCR. The value can be set to True however if the OCR is required on this real text.

PdfToImageForceVectorCheck - This setting is useful when dealing with documents that contains vector objects (e.g. CAD drawings). By default, pages that contain only vector objects are rasterized. Pages that do not have any images but contains vector objects as well as electronic text are skipped from rasterization. However, sometimes there can be a page that contain vector objects (CAD drawings) but its title may be in electronic text. To force rasterizing pages like these, set this property to true.

Quality - JPEG quality setting (percentage value 1 - 100) for use in saving the background and foreground images. Default value is 75

RemoveLines – The value used in Line removal. If blank no line removal will occur. The normal value to use to enable line removal is 100.5 but if you are experience difficulties with this value or have any questions then please contact support@aquaforest.com.

4 Aquaforest Extended OCR Module

4.1 Overview

The Extended OCR module extends the SDK with an additional OCR engine and has the following benefits over and above the standard Aquaforest OCR engine:

- IRIS OCR engine providing enhanced recognition
- Support for 129 languages. See [section 5.4](#) for more details.
- Optional Asian languages support: Standard Chinese, Traditional Chinese, Korean and Japanese.
- Optional Arabic and Farsi language support.
- Support for multiple languages within a single page or document from the same character set.
- Support for multiple document output formats: PDF, DOCX, WORDML, RTF, CSV, XLSX, EXCELML, TXT, HTML, EPUB and XPS
- Multiple PDF version support including PDF A-1a, PDF A-1b, PDF A-2a and PDF A-2b. See [section 5.4](#) for more details.
- Optional Intelligent High Quality Compression

4.2 System Requirements

4.2.1 Supported Environments

- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2016

4.2.2 .NET Framework

.NET Version 4.5.2

4.2.3 Visual C++ Runtime

- Visual C++ 2010 redistributables x86: for 32-bit architectures
- Visual C++ 2010 redistributables x64: for 64-bit architectures

4.3 Folders

The Extended OCR SDK contains the following folders:

- xbin - Contains the binaries used by the Extended OCR module
- xbin/resources - contains all the resources needed for characters recognition, such as lexicons and fonts dictionaries
- docs - contains the documentation of the SDK
- xsamples - contains samples (in C# and VB.NET) illustrating how to make use of the Extended OCR module in common use cases

4.4 Application Development and Deployment

4.4.1 Prerequisites

The table below shows the prerequisites needed for building applications using the Extended OCR engine.

Application Platform	VC++ Redistributable	Minimum .NET Framework Version	Minimum Visual Studio Version
x86	VC ++ 2010 x86	.NET Framework 4.5.2	Visual Studio 2012
x64	VC++ 2010 x64		
Any CPU	VC ++ 2010 x86 and VC++ 2010 x64		

There is a diagnostic tool found at “[SDK installation path]\diagnostics\Aquaforest OCR SDK Prerequisite Check.exe” which can be used to check if the correct versions of .NET Framework and Visual C++ Redistributables are installed.

Note: If you are using Visual Studio 2012 or 2013 to build your application(s), you need to ensure that you also have the [.NET Framework 4.5.2 Multi-targeting pack](#) installed so as to be able to build applications that target .NET Framework 4.5.2.

4.4.2 References

A reference to the Aquaforest.ExtendedOCR.Api dll should be added in your application. If you wish to access the results of the OCR on a word by word basis, for example to obtain word and character results including positional information then you will also need to reference Aquaforest.ExtendedOCR.Shared dll.

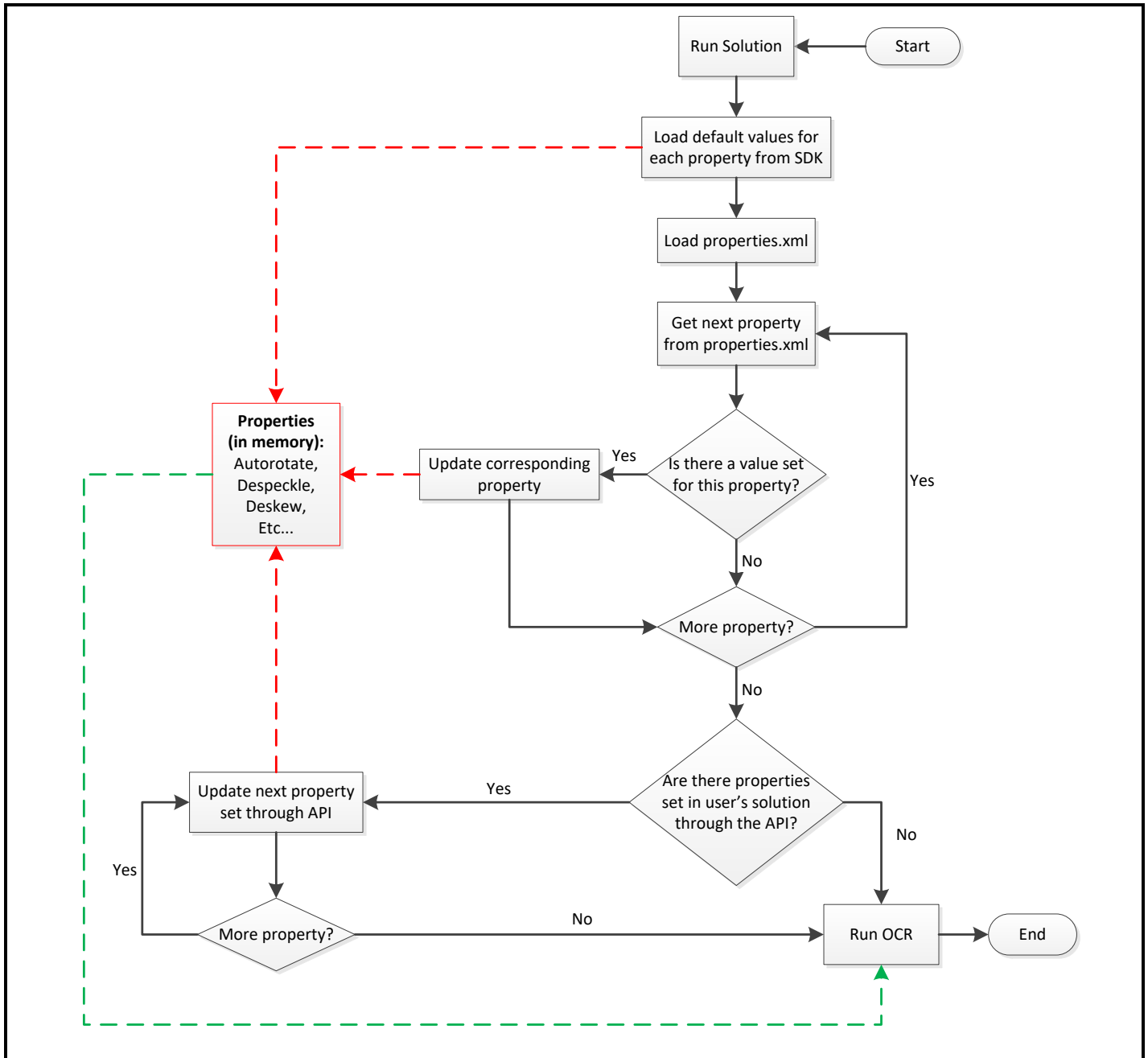
4.4.3 Properties.xml

The properties.xml file located at **bin/properties.xml** contains all the settings provided through the [API](#). Its primary function is to enable users to change pre-processing and OCR settings after the application/solution has been developed.

For instance, if you had particular settings that you did not want to make available through your application for users to change but you still wanted to have the option to configure them in the event there are documents that require special treatment by having additional pre-processing settings to be applied in order to get satisfactory results, then the properties.xml might be useful.

However, for this to work, those settings should not be set through the API, i.e., in your code. This is because settings that are set through the API take precedence over settings set in the properties.xml.

This is depicted in the flowchart below:



If no properties are set through the API, then the SDK will use the values set through the properties .xml.

4.4.4 Deploying C# and VB.NET Applications

Ensure that the target system meets the System Requirements described in [section 4.2](#). Once, the target environment is set up, copy you solution/application files as well as the full contents of the SDK **xbin** folder to the target environment. Make sure the resource folder specified in the OCR constructor inside your solution/application is set to the **resource** folder inside the bin folder you copied to the target machine. For instance:

```
Ocr ocr = new Ocr(@"C:\TargetMachine\MyApp\bin\resources");
```

The SDK contains an in-built functionality to detect if all the required assemblies and files required by the SDK are present. If they are not, an exception will be thrown listing all the files that are missing.

5 Aquaforest Extended OCR Module API Reference

5.1 PreProcessor Class

The PreProcessor class manages all the pre-processing settings available to manipulate the input image before it is passed for Optical Character Recognition. Applying pre-processing settings to low quality source images can improve the quality of OCR.

5.1.1 Constructor

```
PreProcessor preProcessor = new PreProcessor();
```

5.1.2 Properties

Refer to the **OCRSDK 2.2 (Extended engine).chm** file found in the folder "[SDK installation path]\docs\help" to view all the properties available in the PreProcessor class.

5.2 Ocr Class

The OCR object is used to control OCR processing, obtain status updates during processing and retrieve the resulting output from processing upon completion.

5.2.1 Constructor

```
Ocr ocr = new Ocr(@"C:\Aquaforest\OCRSDK\xbin\resources");
```

5.2.2 Properties and Methods

Refer to the **OCRSDK 2.2 (Extended engine).chm** file found in the folder "[SDK installation path]\docs\help" to view all the properties and methods available in the Ocr class.

5.2.3 Events

Event	Description
void StatusUpdate (Object sender, StatusUpdateEventArgs statusUpdateEventArgs)	This event is raised when processing of a page is complete. The StatusUpdateEventArgs object provides access to information relating to the status of the page processed.

5.2.4 Subscribing to StatusUpdate

```
using Aquaforest.ExtendedOCR.Api;
using Aquaforest.ExtendedOCR.Shared;
using System;
using System.IO;

namespace GetTextFromPage
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                using (Ocr ocr = new Ocr(@"C:\Aquaforest\OCRSDK\xbin\resources"))
                {
                    ocr.EnableConsoleOutput = true;
                    ocr.Language = SupportedLanguages.English;
                }
            }
        }
    }
}
```

```

ocr.StatusUpdate += new Ocr.StatusUpdateEventHandler(OcrStatusUpdate);

PreProcessor preProcessor = new PreProcessor();
preProcessor.Deskew = true;
preProcessor.Autorotate = true;

ocr.ReadTIFFSource(Path.GetFullPath(@"..\..\..\documents\source\sample.tif"));

ocr.Recognize(preProcessor);

ocr.DeleteTemporaryFiles();
    }
}
catch (Exception e)
{
    Console.WriteLine("Error in OCR Processing : " + e.Message);
}
}

static void OcrStatusUpdate(object sender, StatusUpdateEventArgs pageCompletedEventArgs)
{
    bool isPageBlank = pageCompletedEventArgs.BlankPage;
    bool isImageAvailable = pageCompletedEventArgs.ImageAvailable;
    bool isTextAvailable = pageCompletedEventArgs.TextAvailable;
    int currentPageNumber = pageCompletedEventArgs.PageNumber;
    int rotation = pageCompletedEventArgs.Rotation;
}
}
}

```

5.3 StatusUpdateEventArgs Class

This class contains information relating to the conversion status of a page.

5.3.1 Constructor

An instance of this class is obtained for each page processed when subscribing to the [StatusUpdate](#) event.

5.3.2 Properties

Property	Description
bool BlankPage	Indicates whether the page was detected as blank.
bool ImageAvailable	Indicates whether an image was successfully extracted (after applying all the appropriate pre-processing settings).
int PageNumber	Returns page for which the object relates to.
int Rotation	The rotation in Degrees (°) of the current page. If Autorotate is set to false this will always be 0.
bool TextAvailable	Indicates whether text was extracted for the page.

5.3.3 Words Class

This class contains a collection of [WordData](#) objects which are available on a page by page basis.

5.3.4 Words Constructor

An instance of this class is obtained by calling the [ReadPageWords](#) method on the `Ocr` object, passing the page for which the words are required.

5.3.5 Words Properties

Property	Description
<code>int Count</code>	Returns the number of WordData objects in the collection.
<code>int Height</code>	Returns the height of the current word.
<code>int Width</code>	Returns the width of the current word.

5.3.6 Words Methods

Method	Description
<code>WordData GetFirst()</code>	Returns the first WordData object in the collection and sets the index to this item.
<code>WordData GetNext()</code>	Returns the next WordData object in the collection and sets the index to this item.
<code>int GetHeight(int index)</code>	Returns the word height from the WordData object stored at the specified index in the collection.
<code>int GetWidth(int index)</code>	Returns the word width from the WordData object stored at the specified index in the collection.

5.3.7 WordData Class

This class contains the individual characters along with the positional information relating to each character in the word and to the word as a whole.

5.3.8 WordData Properties

Property	Description
<code>int Bottom</code>	Gets the Y-coordinate of the bottom edge of the word in pixels.
<code>List<CharacterData> CharacterList</code>	Gets the list of characters in the word.
<code>int Height</code>	Gets the height of the word in pixels.
<code>int Left</code>	Gets the X-coordinate of the left edge of the word in pixels.
<code>int Top</code>	Gets the Y-coordinate of the top edge of the word in pixels.
<code>int Width</code>	Gets the width of the word in pixels.
<code>string Word</code>	Gets the string representation of the word.

5.3.9 CharacterData Class

The character class contains information describing a single character extracted from the Extended OCR engine.

5.3.10 CharacterData Properties

Property	Description
Baseline	Gets the Y-coordinate of the bottom edge of the character in pixels.
Character	Gets the string representation of the character.
Height	Gets the height of the character in pixels.
Width	Gets the width of the character in pixels.
X	Gets the X-coordinate of the left edge of the character in pixels.
Y	Gets the Y-coordinate of the top edge of the character in pixels.

5.4 Supported Languages

Extended OCR accepts up to 8 recognition languages at a time. This is helpful to process mixed documents but, because of the various character sets, not all combinations are allowed. For this reason, the multiple languages support is limited to a single alphabet. For example, Russian and French can't be mixed.

Note:

- Asian languages can't be mixed.
- Extended OCR SDK cannot load more than one Asian language at a time.

Name	Code	Description
English	0	English (American)
German	1	
French	2	
Spanish	3	
Italian	4	
British	5	
Swedish	6	
Danish	7	
Norwegian	8	
Dutch	9	
Portuguese	10	
Brazilian	11	
Galician	12	
Icelandic	13	
Greek	14	
Czech	15	
Hungarian	16	
Polish	17	
Romanian	18	

Name	Code	Description
Slovak	19	
Croatian	20	
Serbian	21	
Slovenian	22	
Luxemb	23	
Finnish	24	
Turkish	25	
Russian	26	
Byelorussian	27	
Ukrainian	28	
Macedonian	29	
Bulgarian	30	
Estonian	31	
Lithuanian	32	
Afrikaans	33	
Albanian	34	
Catalan	35	
Irish_Gaelic	36	
Scottish_Gaelic	37	
Basque	38	
Breton	39	
Corsican	40	
Frisian	41	
Nynorsk	42	
Indonesian	43	
Malay	44	
Swahili	45	
Tagalog	46	
Japanese	47	You will need to have the Asian OCR license to use this language.
Korean	48	You will need to have the Asian OCR license to use this language.
Schinese	49	You will need to have the Asian OCR license to use this language.
Tchinese	50	You will need to have the Asian OCR license

Name	Code	Description
		to use this language.
Quecha	51	
Aymara	52	
Faroese	53	
Friulian	54	
Greenlandic	55	
Haitian_Creole	56	
Rhaeto_Roman	57	
Sardinian	58	
Kurdish	59	
Cebuano	60	
Bemba	61	
Chamorro	62	
Fijan	63	
Ganda	64	
Hani	65	
Ido	66	
Interlingua	67	
Kicongo	68	
Kinyarwanda	69	
Malagasy	70	
Maori	71	
Mayan	72	
Minangkabau	73	
Nahuatl	74	
Nyanja	75	
Rundi	76	
Samoan	77	
Shona	78	
Somali	79	
Sotho	80	
Sundanese	81	
Tahitian	82	
Tonga	83	

Name	Code	Description
Tswana	84	
Wolof	85	
Xhosa	86	
Zapotec	87	
Javanese	88	
Pidgin_Nigeria	89	
Occitan	90	
Manx	91	
Tok_Pisin	92	
Bislama	93	
Hiligaynon	94	
Kapampangan	95	
Balinese	96	
Bikol	97	
Ilocano	98	
Madurese	99	
Waray	100	
None	101	No language, Latin alphabet
Serbian_Latin	102	
Latin	103	
Latvian	104	
Numeric	114	This language limits recognition to numeric characters.
Esperanto	115	
Maltese	116	
Zulu	117	
Afaan	118	
Asturian	119	
AzeriLatin	120	
Luba	121	
Papamientto	122	
Tatar	123	
Turkmen	124	
Welsh	125	

Name	Code	Description
Arabic	126	You will need to have the Arabic OCR license to use this language. Note: Works only for arabic texts with embedded english words. The result for a zone with only English will be empty.
Farsi	127	You will need to have the Arabic OCR license to use this language. Note: Works only for arabic texts with embedded english words. The result for a zone with only English will be empty.
Mexican	128	
BosnianLatin	129	Bosnian (Latin). CharsetCategory.E
BosnianCyrillic	130	Bosnian (Cyrillic). CharsetCategory.D
Moldovan	131	Moldovan. CharsetCategory.E
SwissGerman	132	German (Switzerland). CharsetCategory.C
Tetum	133	Tetum. CharsetCategory.C
Kazakh	134	Kazakh (Cyrillic). CharsetCategory.D
MongolianCyrillic	135	Mongolian (Cyrillic). CharsetCategory.D
UzbekLatin	136	Uzbek (Latin). CharsetCategory.C

6 Aquaforest OCR vs. Extended OCR

6.1 Differences between Aquaforest OCR and Extended OCR

6.1.1 References

Aquaforest	Extended
Aquaforest.OCR.Api	Aquaforest.ExtendedOCR.Api
Aquaforest.OCR.Definitions	Aquaforest.ExtendedOCR.Shared

6.1.2 Ocr Methods

Aquaforest	Extended
AppendPDFOutputToMerger	n/a
DeleteTemporaryFilesForPage	n/a
<code>Ocr ocr = new Ocr();</code>	<code>Ocr ocr = new Ocr(@"C:\Aquaforest\OCRSDK\xbin\resources");</code>
n/a	ReadJPEGSource
n/a	SaveCSVOutput
n/a	SaveDOCXOutput
n/a	SaveExcelMLOutput
n/a	SaveHTMLOutput
n/a	SaveOpenDocumentTextOutput
SavePDFAOutput	Set through PDFVersion: <code>ocr.PDFVersion = PDFVersion.PDF_1_4_Ab</code>
n/a	SaveWordMLOutput
n/a	SaveXLSXOutput
n/a	SaveXPSOutput

6.1.3 Ocr Properties

Aquaforest	Extended
AdvancedPreProcessing / OptimiseOcr	n/a
ConvertToTiff = true	ExtractImageMethod = ExtractImageMethod.ConvertToTiff
CreateProcess	n/a
DeleteTemporaryFilesOnPageCompletion	n/a
DictionaryLookup	n/a
n/a	EmbedFonts
n/a	EnableCsvOutput
n/a	EnableDocxOutput
n/a	EnableExcelMLOutput

Aquaforest	Extended
n/a	EnableHtmlOutput
n/a	EnableOpenDocumentTextOutput
n/a	EnableWordMIOutput
n/a	EnableXlsxOutput
n/a	EnableXpsOutput
int EnableDebugOutput	bool EnableDebugOutput
ErrorMode	n/a
FlipDetect	n/a
n/a	GetPdfTextTimeout
Heuristics	n/a
n/a	IHQCCompression
n/a	Languages (more than one language)
n/a	Layout
MrcTimeout	n/a
OcrProcessSetupTimeout	n/a
OcrTimeout	ProcessPageTimeout
n/a	PdfToImageExtractionTimeout
n/a	PDFVersion
PipeClientConnectionTimeout	n/a
ResourceFolder	This is now set when instantiating the Ocr class
RestartEngineEvery	n/a
n/a	ThreadCount
UseAquaforestImagingFontSizing	n/a
WordMatchThreshold	n/a
n/a	WorkDepth

6.1.4 PreProcessor Properties

Aquaforest	Extended
n/a	AdvancedDespeckle
Binarize	Binarization
BlackPixelLimit	n/a
n/a	BlankPageRemoval
BlankPageThreshold	BlankPageRemoval.Sensitivity
BoxSize	n/a

Aquaforest	Extended
Dotmatrix	n/a
n/a	Dpi
GrayscaleQuality	n/a
n/a	Interpolation
Jbig2EncFlags	n/a
LibTiffSavePageAsBmp	n/a
MaxDeskew	n/a
MinDeskewConfidence	n/a
Morph	n/a
Mrc	n/a
MRCQuality	n/a
MRCBackgroundFactor	n/a
MRCForegroundFactor	n/a
n/a	RemoveDarkBorders
RemoveLines	LineRemoval.RemoveLines
SavePredespeckle	KeepOriginalImage
n/a	NonImagePDF
OneColumn	n/a
Tables	n/a
TextLayerMaxBoxes	n/a
TextLayerFilterHeight	n/a
TextLayerFilterHeightInverted	n/a
TextLayerFilterPercentage	n/a
TextLayerFilterPercentageInverted	n/a
TextLayerFilterRatio	n/a
TextLayerFilterRatioInverted	n/a
TextLayerFilterWidth	n/a
TextLayerFilterWidthInverted	n/a

6.2 Creating a simple application

This section demonstrates how to create a simple application using the SDK.

The simple application is going to convert a TIFF file into a searchable PDF document. It will also need to be able to:

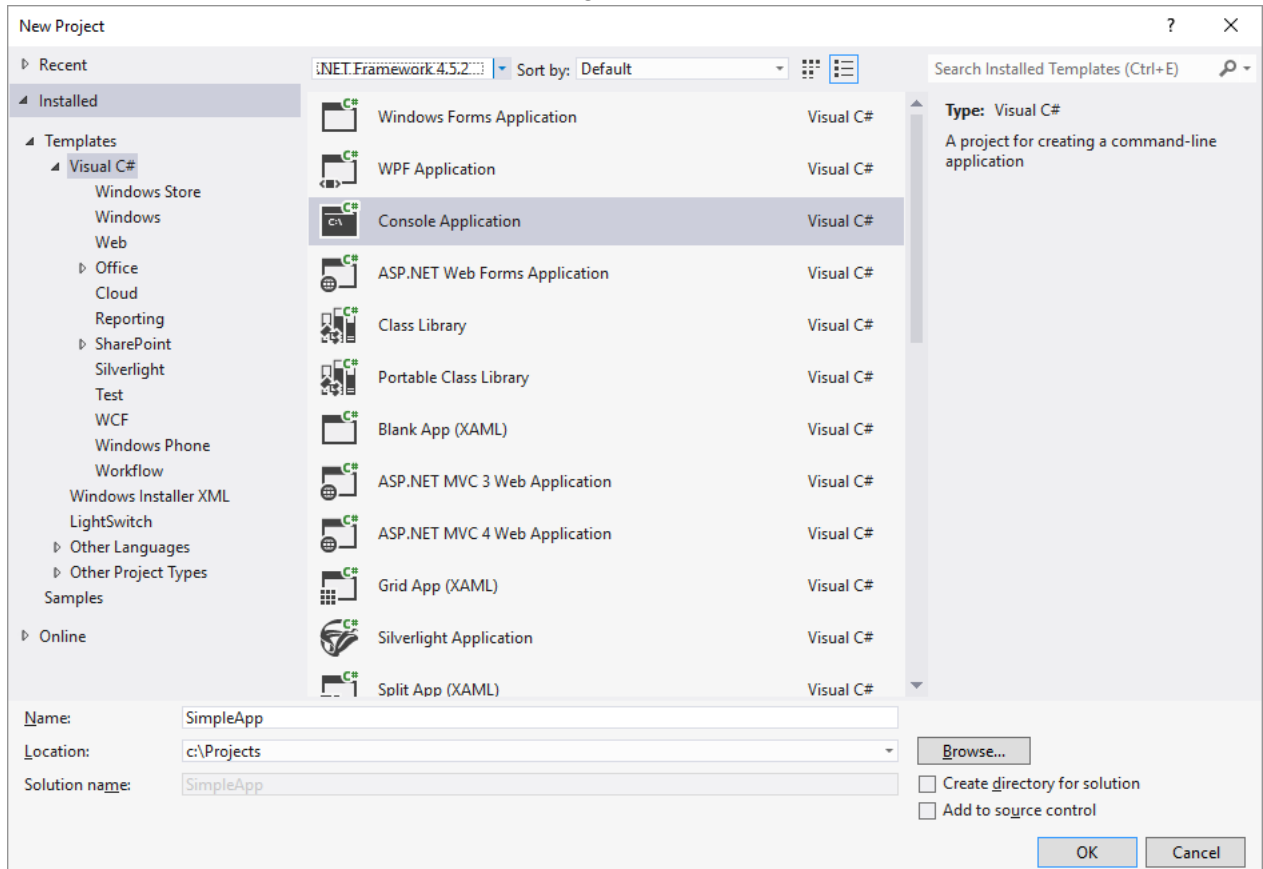
- display the rotation of each page
- identify whether each page contains text or not
- recognize text in different languages

The source TIFF file that will be used contains 6 pages.

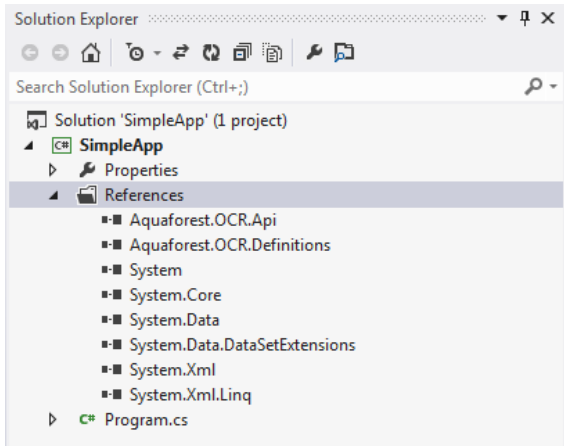
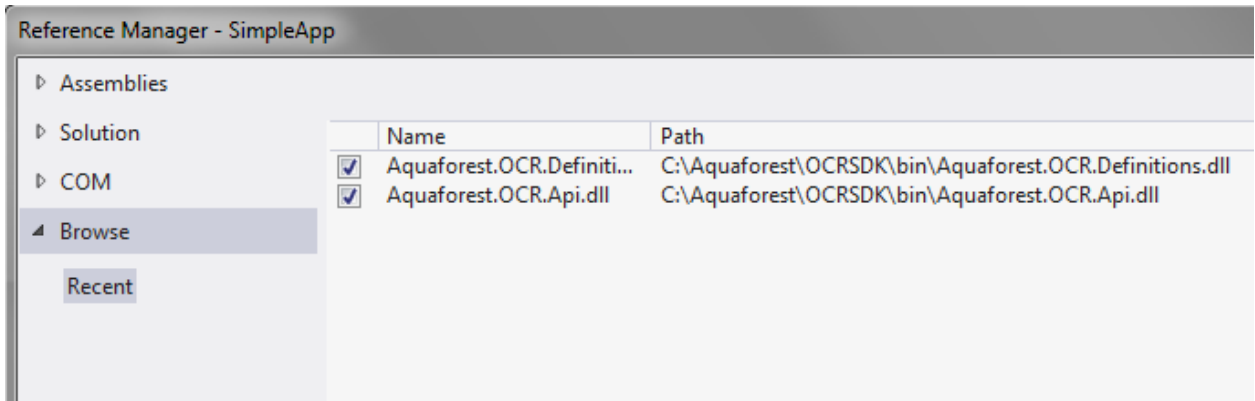
- Page 1 and 4 contains text in English
- Page 2 contains no text
- Page 3 contains text in 4 different languages: English, French, Spanish and German
- Page 5 is rotated 180°
- Page 6 contains text in French

6.2.1 Using Aquaforest SDK and Visual Studio 2012+

1. Create a new console application, specifying .NET Framework 4.5.2.



2. Add a reference to Aquaforest.OCR.Api.dll and Aquaforest.OCR.Definitions.dll.



3. Open Program.cs and add the following "using" directives:

```
using Aquaforest.OCR.Api;  
using Aquaforest.OCR.Definitions;
```

This will result in the following code:

```
using Aquaforest.OCR.Api;  
using Aquaforest.OCR.Definitions;  
using System;  
  
namespace SimpleApp  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
        }  
    }  
}
```

4. Next, add the code to convert the TIFF document to a searchable PDF document.

```
using Aquaforest.OCR.Api;
using Aquaforest.OCR.Definitions;
using System;

namespace SimpleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            using (Ocr ocr = new Ocr())
            {
                string resourceFolder = @"C:\Aquaforest\OCRSdk\bin";

                string currentEnvironmentVariables =
                    Environment.GetEnvironmentVariable("PATH");
                if (!currentEnvironmentVariables.Contains(resourceFolder))
                {
                    Environment.SetEnvironmentVariable("PATH",
                        currentEnvironmentVariables + ";" + resourceFolder);
                }

                ocr.ResourceFolder = resourceFolder;
                ocr.EnableConsoleOutput = true;
                ocr.EnablePdfOutput = true;
                ocr.Language = SupportedLanguages.English;

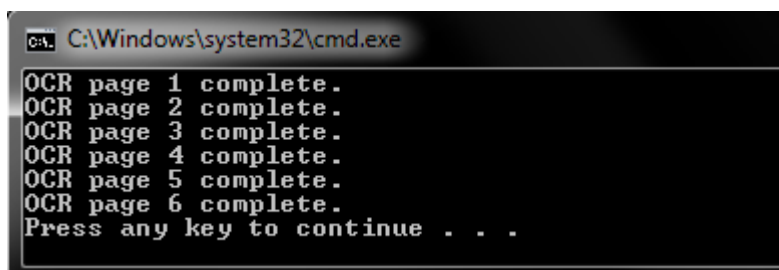
                PreProcessor preProcessor = new PreProcessor();
                preProcessor.Autorotate = true;
                preProcessor.Deskew = true;

                ocr.ReadTIFFSource(@"C:\MyFiles\input\sample.tif");

                if (ocr.Recognize(preProcessor))
                {
                    ocr.SavePDFOutput(@"C:\MyFiles\output\sample.pdf", true);
                }

                ocr.DeleteTemporaryFiles();
            }
        }
    }
}
```

If you run the above code, it will give the following output in the console and generate a searchable PDF file:



```
C:\Windows\system32\cmd.exe
OCR page 1 complete.
OCR page 2 complete.
OCR page 3 complete.
OCR page 4 complete.
OCR page 5 complete.
OCR page 6 complete.
Press any key to continue . . .
```

5. Now, add the code to display the following details about each page:

- Rotation
- Whether or not it contains text

The codes that have been added are highlighted below:

```
using Aquaforest.OCR.Api;
using Aquaforest.OCR.Definitions;
using System;

namespace SimpleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            using (Ocr ocr = new Ocr())
            {
                string resourceFolder = @"C:\Aquaforest\OCRSdk\bin";

                string currentEnvironmentVariables =
                    Environment.GetEnvironmentVariable("PATH");
                if (!currentEnvironmentVariables.Contains(resourceFolder))
                {
                    Environment.SetEnvironmentVariable("PATH",
                        currentEnvironmentVariables + ";" + resourceFolder);
                }

                ocr.ResourceFolder = resourceFolder;
                ocr.EnableConsoleOutput = true;
                ocr.EnablePdfOutput = true;
                ocr.Language = SupportedLanguages.English;
                ocr.StatusUpdate += OcrStatusUpdate;

                PreProcessor preProcessor = new PreProcessor();
                preProcessor.Autorotate = true;
                preProcessor.Deskew = true;

                ocr.ReadTIFFSource(@"C:\MyFiles\input\sample.tif");

                if (ocr.Recognize(preProcessor))
                {
                    ocr.SavePDFOutput(@"C:\MyFiles\output\sample.pdf", true);
                }

                ocr.DeleteTemporaryFiles();
            }
        }

        static void OcrStatusUpdate(object sender, StatusUpdateEventArgs pageCompleteEventArgs)
        {
            Console.WriteLine(new string('-', 50));
            Console.WriteLine("Page {0}", pageCompleteEventArgs.PageNumber);
            Console.WriteLine("Contains Text: {0}", pageCompleteEventArgs.TextAvailable);
            Console.WriteLine("Rotation: {0}", pageCompleteEventArgs.Rotation);
        }
    }
}
```

If you run the above code, you will get the following console output:

```
cs. C:\Windows\system32\cmd.exe
-----
Page 1
Contains Text: True
Rotation: 0
OCR page 1 complete.
-----
Page 2
Contains Text: False
Rotation: 0
-----
Page 3
Contains Text: True
Rotation: 0
OCR page 3 complete.
-----
Page 4
Contains Text: True
Rotation: 0
OCR page 4 complete.
-----
Page 5
Contains Text: True
Rotation: 2
OCR page 5 complete.
-----
Page 6
Contains Text: True
Rotation: 0
OCR page 6 complete.
Press any key to continue . . .
```

As mentioned previously, page 2 contains no text and page 5 is rotated 180°. This is clearly shown in the above output. The value of rotation for Page 5 is 2 because rotation is described in 90° steps from beginning orientation (0), i.e. 1 = 90, 2 = 180, 3 = 270.

So far, the application can display the rotation of each and identify whether each page contains text or not. However, it is only recognizing text using the English language since we set the language to English:

```
ocr.Language = SupportedLanguages.English;
```

Consequently, the SDK is not recognizing the 4 languages on page 3 and the French language on page 6. The results of these 2 pages are shown below. The text has been copied from the resulting searchable PDF file.

Page 3 Image:

EN	FR	ES	DE								
<p>Using the Remote</p> <p>Use the remote to control compatible media phone players.</p> <table border="1"> <tr> <td>Answer/End Call</td> <td>Click the center button once.</td> </tr> </table> <p>When answering calls speak in a normal manner.</p> <p>Answer/end calls on most phones by pressing the center button once.</p> <p>Please read your mobile device's user guide for more information on how to use the answer/end button, check for additional features or to troubleshoot usage problems.</p> <p>NOTE: On some phones it might be necessary to adjust the volume when switching from phone call to music.</p> <p>For more information about compatible models go to: www.shure.com</p>	Answer/End Call	Click the center button once.	<p>Utilisation de la télécommande</p> <p>Utiliser la télécommande pour commander les lecteurs multimédia sur téléphones compatibles.</p> <table border="1"> <tr> <td>Réponse / Fin d'appel</td> <td>Cliquer une fois sur le bouton central</td> </tr> </table> <p>Parler d'une voix normale pour répondre aux coups de téléphone.</p> <p>Répondre à / terminer les appels sur la plupart des téléphones en appuyant une fois sur le bouton central.</p> <p>Prière de lire le guide d'utilisation de l'appareil mobile pour trouver de plus amples renseignements sur l'emploi du bouton Réponse / Fin d'appel, pour connaître les fonctions supplémentaires ou pour dépanner les problèmes d'utilisation.</p> <p>REMARQUE : Sur certains téléphones il peut s'avérer nécessaire de régler le volume lorsqu'on passe des coups de téléphone à la musique.</p> <p>Pour de plus amples renseignements sur les modèles compatibles, visiter : www.shure.com</p>	Réponse / Fin d'appel	Cliquer une fois sur le bouton central	<p>Uso del control remoto</p> <p>Usate il telecomando per agire sul vostro lettore multimediale compatibile.</p> <table border="1"> <tr> <td>Invio/ Fine chiamata</td> <td>Pulse el botón central una vez</td> </tr> </table> <p>Quando rispondete alle chiamate, parlate normalmente.</p> <p>I comandi Invio/Fine chiamata sulla maggior parte di telefoni vengono effettuati mediante pressione del pulsante centrale. Per ulteriori informazioni sull'uso del pulsante di invio/fine chiamata, sulla ricerca di funzioni aggiuntive o sull'individuazione di problemi d'uso, leggete la Guida utente del vostro dispositivo mobile.</p> <p>NOTA – è possibile che su alcuni telefoni sia necessario regolare il volume quando si passa dalla telefonata all'ascolto di musica.</p> <p>Para más información sobre modelos compatibles, acuda a: www.shure.com</p>	Invio/ Fine chiamata	Pulse el botón central una vez	<p>Verwendung der Fernsteuerung</p> <p>Die Fernsteuerung zur Bedienung kompatibler Medienwiedergabegeräte/ Handys verwenden.</p> <table border="1"> <tr> <td>Anruf annehmen/ beenden</td> <td>Mittlere Taste einmal anklicken.</td> </tr> </table> <p>Beim Telefonieren auf normale Weise sprechen.</p> <p>Bei den meisten Handys werden Anrufe angenommen/beendet, indem die mittlere Taste einmal gedrückt wird.</p> <p>In der Bedienungsanleitung Ihres Handys finden Sie weitere Informationen über die Verwendung der Taste Annehmen/Beenden sowie über sonstige technische Eigenschaften oder die Störungssuche bei Nutzungsproblemen.</p> <p>HINWEIS: Bei manchen Telefonen ist es eventuell nötig, die Lautstärke anzupassen, wenn von einem Telefongespräch auf Musik umgeschaltet wird.</p> <p>Weitere Informationen über kompatible Modelle sind auf unserer Website zu finden: www.shure.com</p>	Anruf annehmen/ beenden	Mittlere Taste einmal anklicken.
Answer/End Call	Click the center button once.										
Réponse / Fin d'appel	Cliquer une fois sur le bouton central										
Invio/ Fine chiamata	Pulse el botón central una vez										
Anruf annehmen/ beenden	Mittlere Taste einmal anklicken.										

Page 3 OCR Results:

<p>Using the Remote</p> <p>Use the remote to control compatible media phone players.</p> <p>Answer/End Call</p> <p>Click the center button once.</p> <p>When answering calls speak in a normal manner.</p> <p>Answer/end calls on most phones by pressing the center button once.</p> <p>Please read your mobile device's user guide for more information on how to use the answer/end button, check for additional features or to troubleshoot usage problems.</p> <p>NOTE: On some phones it might be necessary to adjust the volume when switching from phone call to music.</p> <p>For more information about compatible models go to: www.shure.com</p>	<p>Utilisation de la télécommande</p> <p>Utiliser la télécommande pour commander les lecteurs multimedia sur telephones compatibles.</p> <p>Réponse / Fin d'appel</p> <p>Cliquer une fois sur le bouton central</p> <p>Parler d'une voix normale pour répondre aux coups de telephone.</p> <p>Repondre a / terminer les appels sur la plupart des telephones en appuyant une fois sur le bouton central.</p> <p>Pour de plus amples renseignements sur les modeles compatibles, visiter: www.shure.com</p> <p>Prière de lire le guide d'utilisation de l'appareil mobile pour trouver de plus amples renseignements sur l'emploi du bouton Reponse / Fin d'appel, pour connaître les fonctions supplementaires ou pour depanner les problemes d'utilisation.</p> <p>REMARQUE: Sur certains telephones il peut s'averer necessaire de regler le volume lorsqu'on passe des coups de telephone a la musique.</p>	<p>Uso del control remoto</p> <p>Usate il telecomando per agire sul vostro lettore multimediale compatibile.</p> <p>Invio/ Fine chiamata</p> <p>Pulse el boton central una vez</p> <p>Quando rispondete alle chiamate, parlate normalmente.</p> <p>Para mas informacion sobre modelos compatibles, acuda a: www.shure.com</p> <p>I comandi Invio/Fine chiamata sulla maggior parte di telefoni vengono effettuati mediante pressione del pulsante centrale. Per ulteriori informazioni sull'uso del pulsante di invio/fine chiamata, sulla ricerca di funzioni aggiuntive o sull'individuazione di problemi d'uso, leggete la Guida utente del vostro dispositivo mobile.</p> <p>NOTA — è possibile che su alcuni telefoni sia necessario regolare il volume quando si passa dalla telefonata all'ascolto di musica.</p>	<p>Verwendung der Fernsteuerung</p> <p>Die Fernsteuerung zur Bedienung kompatibler Medienwiedergabegeräte/ Handys verwenden.</p> <p>Anruf annehmen/ beenden</p> <p>Mittlere Taste einmal anklicken.</p> <p>Beim Telefonieren auf normale Weise sprechen.</p> <p>Bei den meisten Handys werden Anrufe angenommen/beendet, indem die mittlere Taste einmal gedrückt wird.</p> <p>In der Bedienungsanleitung Ihres Handys finden Sie weitere Informationen über die Verwendung der Taste Annehmen/Beenden sowie über sonstige technische Eigenschaften oder die Störungssuche bei Nutzungsproblemen.</p> <p>HINWEIS: Bei manchen Telefonen ist es eventuell nötig, die Lautstärke anzupassen, wenn von einem Telefongespräch auf Musik umgeschaltet wird.</p> <p>Weitere Informationen über kompatible Modelle sind auf unserer Website zu finden: www.shure.com</p>
--	--	--	---

Editeurs de logiciel indépendants

Les éditeurs de logiciels indépendants utilisent HATS Toolkit pour créer des applications personnalisées qui sont ensuite revendues à d'autres clients.

Accessibilité dans HATS

Les fonctions d'accessibilité permettent à un utilisateur souffrant d'un handicap physique tel qu'une mobilité réduite, un trouble de la vision, etc., d'utiliser les logiciels de manière satisfaisante. Etant donné qu'il constitue un ensemble de modules d'extension de Rational SDP, HATS bénéficie des fonctions d'accessibilité fournies par Rational SDP. Les principales fonctions d'accessibilité de Rational SDP sont les suivantes :

- Rational SDP utilise les API Microsoft Active Accessibility (MSAA) pour rendre les éléments de l'interface utilisateur accessibles à la technologie dédiée à l'assistance.
- Vous pouvez activer toutes les fonctions à partir du clavier au lieu d'utiliser la souris.

Remarque : Sur certains systèmes, il se peut que les traits de soulignement des touches de raccourci n'apparaissent pas dans la page des paramètres du composant Sous-fichier. Cette page est accessible à partir de **Paramètres de projet > Rendu > Composants > Sous-fichier > Paramètres**. Si c'est le cas sur votre système, pour afficher tous les traits de soulignement, utilisez les touches **Alt+s** pour accéder à la page, au lieu de cliquer sur le bouton **Paramètres**.

- Vous pouvez utiliser un logiciel de lecteur d'écran tel que JAWS (Job Access With Speech) de Freedom Scientific et un synthétiseur de voix numérique pour reconnaître à l'ouïe ce qui s'affiche à l'écran.
- Vous pouvez grossir l'affichage des vues graphiques.
- Les polices ou couleurs définies par Rational SDP peuvent être configurées dans une boîte de dialogue à laquelle vous accédez en sélectionnant **Fenêtre > Préférences > Informations générales > Présentation > Couleurs et polices**.

Page 6 OCR Results:

Editeurs de logiciel indépendants

Les éditeurs de logiciels indépendants utilisent HATS Toolkit pour créer des applications personnalisées qui sont ensuite revendues à d'autres clients.

Accessibilité dans HATS

Les fonctions d'accessibilité permettent à un utilisateur souffrant d'un handicap physique tel qu'une mobilité réduite, un trouble de la vision, etc., d'utiliser les logiciels de manière satisfaisante. Etant donné qu'il constitue un ensemble de modules d'extension de Rational SDP, HATS bénéficie des fonctions d'accessibilité fournies par Rational SDP. Les principales fonctions d'accessibilité de Rational SDP sont les suivantes:

Rational SDP utilise les API Microsoft Active Accessibility (MSAA) pour rendre les éléments de l'interface utilisateur accessibles à la technologie dédiée à l'assistance.

Vous pouvez activer toutes les fonctions à partir du clavier au lieu d'utiliser la souris.

Remarque : Sur certains systèmes, il se peut que les traits de soulignement des touches de raccourci n'apparaissent pas dans la page des paramètres du composant Sous-fichier. Cette page est accessible à partir de **Paramètres de projet > Rendu > Composants >**

Sous-fichier > Paramètres. Si c'est le cas sur votre système, pour afficher tous les traits de soulignement, utilisez les touches **Alt+s** pour accéder à la page, au lieu de cliquer sur le bouton **Paramètres**.

Vous pouvez utiliser un logiciel de lecteur d'écran tel que JAWS (Job Access With Speech) de Freedom Scientific et un synthétiseur de voix numérique pour reconnaître à l'ouïe ce qui s'affiche à l'écran.

Vous pouvez grossir l'affichage des vues graphiques.

Les polices ou couleurs définies par Rational SDP peuvent être configurées dans une boîte de dialogue à laquelle vous accédez en sélectionnant **Fenêtre > Préférences > Informations générales > Présentation > Couleurs et polices**.

If we analyse the results of these two pages, we will notice that the SDK did not identify the characters that are specific to the other languages such as ä, á, é, î and ü. For example:

- "telecommande" instead of "télécommande"
- "gedruckt" instead of "gedrückt"
- "Accessibilite" instead of "Accessibilité"
- "reconnaftre" instead of "reconnaître"

To instruct the SDK to recognize text in different languages, we need to use the [advanced pre-processing](#) option.

6. To use the [advanced pre-processing](#) option, we need to modify the properties.xml file and add 3 more "ImagePreProcessing" sections (one for each language).

```
<Properties>
  <ErrorHandling>
    <RestartEngineEvery>10</RestartEngineEvery>
    <ErrorMode>7</ErrorMode>
  </ErrorHandling>
  <ImagePreProcessing ID="1">
    <!--French-->
    <Language>2</Language>
  </ImagePreProcessing>
  <ImagePreProcessing ID="2">
    <!--Spanish-->
    <Language>5</Language>
  </ImagePreProcessing>
  <ImagePreProcessing ID="3">
    <!--German-->
    <Language>1</Language>
  </ImagePreProcessing>
  <ImagePreProcessingDefaults>
    <Binarize>200</Binarize>
    <BlackPixelLimit>0.65</BlackPixelLimit>
    <BoxSize>100</BoxSize>
    <GrayscaleQuality>0</GrayscaleQuality>
    <Jbig2EncFlags>-s</Jbig2EncFlags>
    <Language>0</Language><!--English-->
  </ImagePreProcessingDefaults>
</Properties>
```

The English language is already set in the "ImagePreProcessingDefaults" section.

7. Next, set the AdvancedPreProcessing property of the Ocr object to true in the code. The final code should now look as follows:

```

using Aquaforest.OCR.Api;
using Aquaforest.OCR.Definitions;
using System;

namespace SimpleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            using (Ocr ocr = new Ocr())
            {
                string resourceFolder = @"C:\Aquaforest\OCRSdk\bin";

                string currentEnvironmentVariables =
                    Environment.GetEnvironmentVariable("PATH");
                if (!currentEnvironmentVariables.Contains(resourceFolder))
                {
                    Environment.SetEnvironmentVariable("PATH",
                        currentEnvironmentVariables + ";" + resourceFolder);
                }

                ocr.ResourceFolder = resourceFolder;
                ocr.EnableConsoleOutput = true;
                ocr.EnablePdfOutput = true;
                ocr.Language = SupportedLanguages.English;
                ocr.StatusUpdate += OcrStatusUpdate;
                ocr.AdvancedPreProcessing = true;

                PreProcessor preProcessor = new PreProcessor();
                preProcessor.Autorotate = true;
                preProcessor.Deskew = true;

                ocr.ReadTIFFSource(@"C:\MyFiles\input\sample.tif");

                if (ocr.Recognize(preProcessor))
                {
                    ocr.SavePDFOutput(@"C:\MyFiles\output\sample.pdf", true);
                }

                ocr.DeleteTemporaryFiles();
            }
        }

        static void OcrStatusUpdate(object sender, StatusUpdateEventArgs pageCompleteEventArgs)
        {
            Console.WriteLine(new string('-', 50));
            Console.WriteLine("Page {0}", pageCompleteEventArgs.PageNumber);
            Console.WriteLine("Contains Text: {0}", pageCompleteEventArgs.TextAvailable);
            Console.WriteLine("Rotation: {0}", pageCompleteEventArgs.Rotation);
        }
    }
}

```

By using [advanced pre-processing](#), the SDK will OCR each page 4 times, one for each "ImagePreProcessing" section added in the properties.xml file in addition to the "ImagePreProcessingDefaults" section. Consequently, this will generate 4 OCR results for each page. The SDK will then compare these results using heuristics and dictionary lookup, and determine the optimum result to use.

Below are the results that are produced using the [advanced pre-processing](#) setting:

Page 3 OCR Results [with advanced pre-processing]:

<p>Using the Remote Use the remote to control compatible media phone players. Answer/End Call Click the center button once. When answering calls speak in a normal manner. Answer/end calls on most phones by pressing the center button once. Please read your mobile device's user guide for more information on how to use the answer/end button, check for additional features or to troubleshoot usage problems. NOTE: On some phones it might be necessary to adjust the volume when switching from phone call to music. For more information about compatible models go to: www.shure.com</p>	<p>Utilisation de la telecommande Utiliser la telecommande pour commander les lecteurs multimedia sur telephones compatibles. Reponse / Fin d'appel Cliquer une fois sur le bouton central Parler d'une voix normale pour repondre aux coups de telephone. Repondre a / terminer les appels sur la plupart des telephones en appuyant une fois sur le bouton central. Pour de plus amples renseignements sur les modeles compatibles, visiter: www.shure.com Priere de lire le guide d'utilisation de l'appareil mobile pour trouver de plus amples renseignements sur l'emploi du bouton Reponse / Fin d'appel, pour connaitre les fonctions supplementaires ou pour depanner les problemes d'utilisation. REMARQUE: Sur certains telephones il peut s'averer necessaire de regler le volume lorsqu'on passe des coups de telephone a la musique.</p>	<p>Uso del control remoto Usate il telecomando per agire sul vostro lettore multimediale compatibile. Invio/Fine chiamata Pulse el boton central una vez Quando rispondete alle chiamate, parlate normalmente. Para mas informacion sobre modelos compatibles, acuda a: www.shure.com I comandi Invio/Fine chiamata sulla maggior parte di telefoni vengono effettuati mediante pressione del pulsante centrale. Per ulteriori informazioni sull'uso del pulsante di invio/fine chiamata, sulla ricerca di funzioni aggiuntive o sull'individuazione di problemi d'uso, leggete la Guida utente del vostro dispositivo mobile. NOTA — e possibile che su alcuni telefoni sia necessario regolare il volume quando si passa dalla telefonata all'ascolto di musica.</p>	<p>Verwendung der Fernsteuerung Die Fernsteuerung zur Bedienung kompatibler Medienwiedergabegerate/ Handys verwenden. Anruf Mittlere annehmen/ Taste einmal beenden anklicken. Beim Telefonieren auf normale Weise sprechen. Bei den meisten Handys werden Anrufe angenommen/beendet, indem die mittlere Taste einmal gedruckt wird. In der Bedienungsanleitung Ihres Handys finden Sie weitere Informationen über die Verwendung der Taste Annehmen/Beenden sowie über sonstige technische Eigenschaften oder die Störungssuche bei Nutzungsproblemen. HINWEIS: Bei manchen Telefonen ist es eventuell nötig, die Lautstärke anzupassen, wenn von einem Telefongespräch auf Musik umgeschaltet wird. Weitere Informationen über kompatible Modelle sind auf unserer Website zu finden: www.shure.com</p>
---	---	---	--

Page 6 OCR Results [with advanced pre-processing]:

<p>Editeurs de logiciel indépendants Les éditeurs de logiciels indépendants utilisent HATS Toolkit pour créer des applications personnalisées qui sont ensuite revendues à d'autres clients.</p> <p>Accessibilité dans HATS</p> <p>Les fonctions d'accessibilité permettent à un utilisateur souffrant d'un handicap physique tel qu'une mobilité réduite, un trouble de la vision, etc., d'utiliser les logiciels de manière satisfaisante. Etant donné qu'il constitue un ensemble de modules d'extension de Rational SDP, HATS bénéficie des fonctions d'accessibilité fournies par Rational SDP. Les principales fonctions d'accessibilité de Rational SDP sont les suivantes :</p> <p>Rational SDP utilise les API Microsoft Active Accessibility (MSAA) pour rendre les éléments de l'interface utilisateur accessibles à la technologie dédiée à l'assistance.</p> <p>Vous pouvez activer toutes les fonctions à partir du clavier au lieu d'utiliser la souris.</p> <p>Remarque : Sur certains systèmes, il se peut que les traits de soulignement des touches de raccourci n'apparaissent pas dans la page des paramètres du composant Sous-fichier. Cette page est accessible à partir de Paramètres de projet > Rendu > Composants > Sous-fichier > Paramètres. Si c'est le cas sur votre système, pour afficher tous les traits de soulignement, utilisez les touches Alt+S pour accéder à la page, au lieu de cliquer sur le bouton Paramètres.</p> <p>Vous pouvez utiliser un logiciel de lecteur d'écran tel que JAWS for Access With Speech) de Freedom Scientific et un synthétiseur de voix numérique pour reconnaître à l'ouïe ce qui s'affiche à l'écran.</p> <p>Vous pouvez grossir l'affichage des vues graphiques.</p> <p>Les polices ou couleurs définies par Rational SDP peuvent être configurées dans une boîte de dialogue à laquelle vous accédez en sélectionnant Fenêtre > Préférences > Informations générales > Présentation > Couleurs et polices.</p>
--

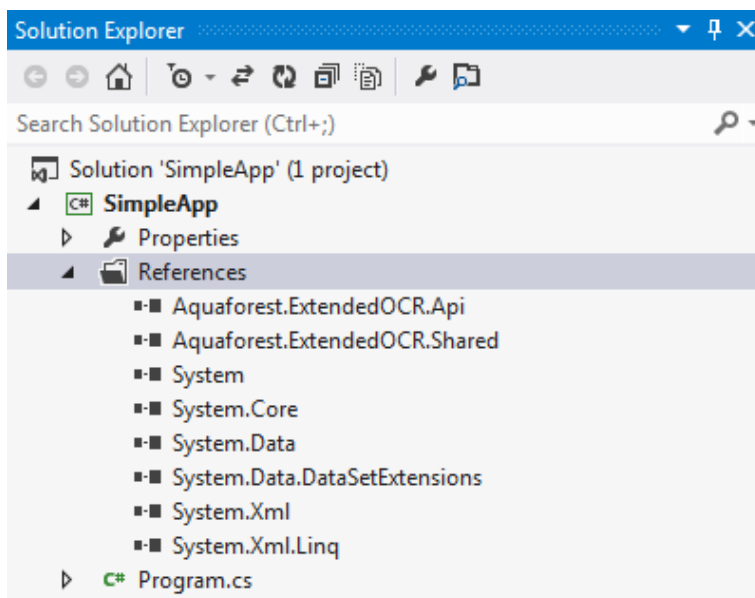
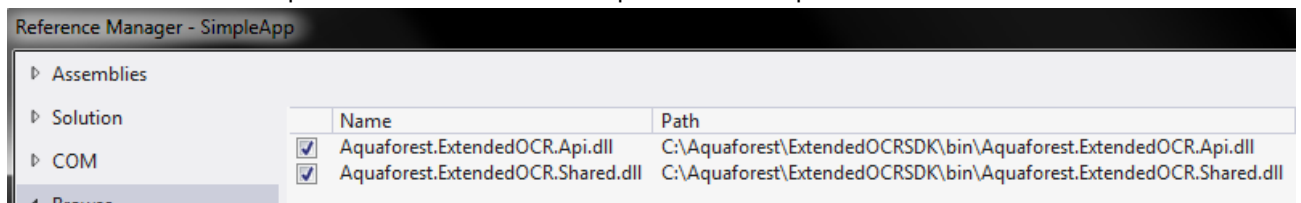
If we analyse the OCR results of page 6, we can see that all the French accents are now recognised properly. However, if we examine the OCR results of page 3, we'll notice that the SDK still did not recognise the special characters of the other languages. It actually produced the same results as the one without advanced pre-processing. The reason for that is the SDK cannot recognise more than one language per page.

In order to overcome this issue, we'll need to use the Extended OCR module. The next section explains how to convert this example to use the Extended OCR module.

6.2.2 Converting to Extended OCR

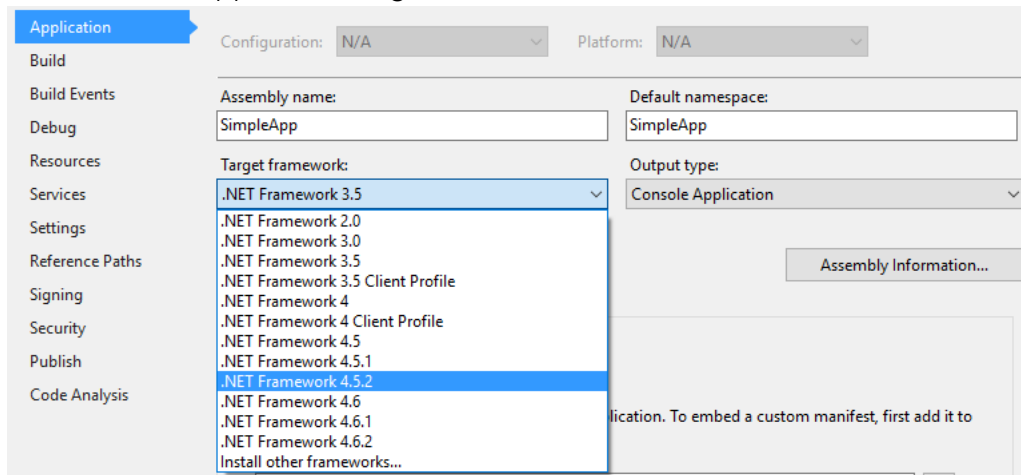
This section explains how to change the application created in the previous section so that it uses the Extended SDK module.

1. Remove the following "using" directives from the code:
`using Aquaforest.OCR.Api;`
`using Aquaforest.OCR.Definitions;`
2. Remove the reference to `Aquaforest.OCR.Api.dll` and `Aquaforest.OCR.Definitions.dll`
3. Add a reference to `Aquaforest.ExtendedOCR.Api.dll` and `Aquaforest.ExtendedOCR.Shared.dll`.



4. Add the following "using" directives in `Program.cs`:
`using Aquaforest.ExtendedOCR.Api;`
`using Aquaforest.ExtendedOCR.Shared;`

5. Make sure the application targets .NET Framework 4.5.2:



6. Next, you will need to modify certain parts of the code. The parts of the code that needs changing are highlighted below:

```
using Aquaforest.OCR.Api;
using Aquaforest.OCR.Definitions;
using System;

namespace SimpleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            using (Ocr ocr = new Ocr())
            {
                string resourceFolder = @"C:\Aquaforest\OCRSdk\bin";

                string currentEnvironmentVariables =
                    Environment.GetEnvironmentVariable("PATH");
                if (!currentEnvironmentVariables.Contains(resourceFolder))
                {
                    Environment.SetEnvironmentVariable("PATH",
                        currentEnvironmentVariables + ";" + resourceFolder);
                }

                ocr.ResourceFolder = resourceFolder;
                ocr.EnableConsoleOutput = true;
                ocr.EnablePdfOutput = true;
                ocr.Language = SupportedLanguages.English;
                ocr.StatusUpdate += OcrStatusUpdate;
                ocr.AdvancedPreProcessing = true;

                PreProcessor preProcessor = new PreProcessor();
                preProcessor.Autorotate = true;
                preProcessor.Deskew = true;

                ocr.ReadTIFFSource(@"C:\MyFiles\input\sample.tif");

                if (ocr.Recognize(preProcessor))
                {
                    ocr.SavePDFOutput(@"C:\MyFiles\output\sample.pdf", true);
                }
                ocr.DeleteTemporaryFiles();
            }
        }

        static void OcrStatusUpdate(object sender, StatusUpdateEventArgs pageCompleteEventArgs)
        {
            Console.WriteLine(new string('-', 50));
            Console.WriteLine("Page {0}", pageCompleteEventArgs.PageNumber);
            Console.WriteLine("Contains Text: {0}", pageCompleteEventArgs.TextAvailable);
            Console.WriteLine("Rotation: {0}", pageCompleteEventArgs.Rotation);
        }
    }
}
```

```

    }
}

```

- Change the `resourcesFolder` location to point the Extended SDK resources folder
- Change `Ocr ocr = new Ocr();` to `Ocr ocr = new Ocr(resourceFolder);`
- Delete `ocr.ResourceFolder = resourceFolder;`
- Delete :

```

string currentEnvironmentVariables = Environment.GetEnvironmentVariable("PATH");
if (!currentEnvironmentVariables.Contains(resourceFolder))
{
    Environment.SetEnvironmentVariable("PATH",
        currentEnvironmentVariables + ";" + resourceFolder);
}

```

- Delete `ocr.AdvancedPreProcessing = true;`
The Extended module does not use `AdvancedPreProcessing` and has a separate `properties.xml` file that does not contain any "ImagePreProcessing" sections. Instead the different languages are set through the "`Languages`" property as described in the next bullet point.
- Delete `ocr.Language = SupportedLanguages.English;` and add the following instead:

```

ocr.Languages = new SupportedLanguages[]
{
    SupportedLanguages.English,
    SupportedLanguages.French,
    SupportedLanguages.Spanish,
    SupportedLanguages.German
};

```

7. The resulting code should look like this:

```

namespace SimpleApp
{
    class Program
    {
        static void Main(string[] args)
        {
            string resourceFolder = @"C:\Aquaforest\OCRSdk\xbin\resources";
            using (Ocr ocr = new Ocr(resourceFolder))
            {
                ocr.EnableConsoleOutput = true;
                ocr.EnablePdfOutput = true;
                ocr.Languages = new SupportedLanguages[]
                {
                    SupportedLanguages.English,
                    SupportedLanguages.French,
                    SupportedLanguages.Spanish,
                    SupportedLanguages.German
                };
                ocr.StatusUpdate += OcrStatusUpdate;

                PreProcessor preProcessor = new PreProcessor();
                preProcessor.Autorotate = true;
                preProcessor.Deskew = true;

                ocr.ReadTIFFSource(@"C:\MyFiles\input\sample.tif");

                if (ocr.Recognize(preProcessor))
                {
                    ocr.SavePDFOutput(@"C:\MyFiles\output\sample.pdf", true);
                }

                ocr.DeleteTemporaryFiles();
            }
        }
    }
}

```

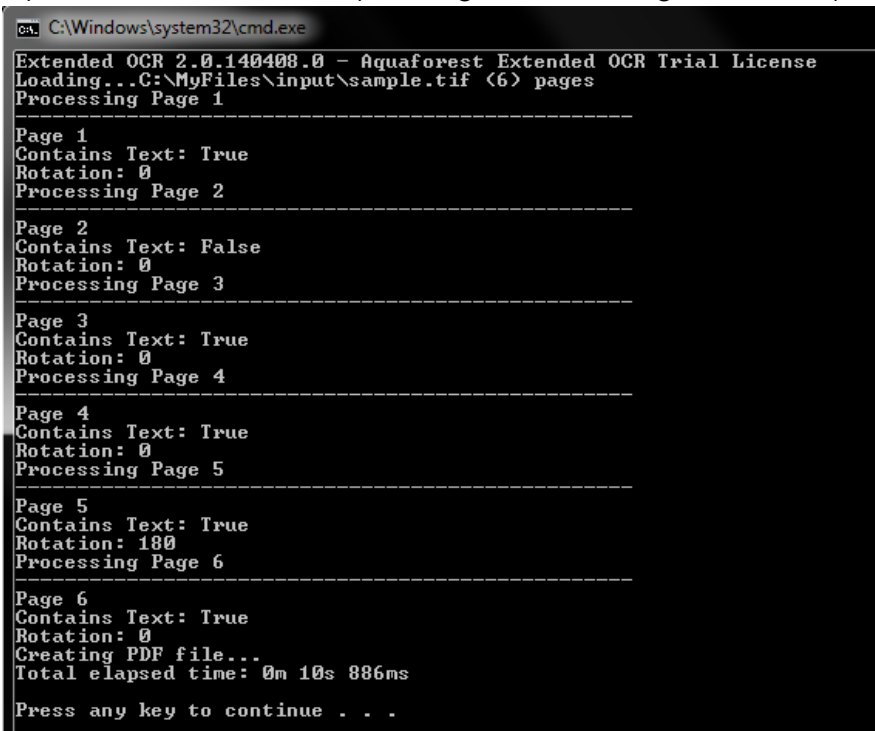


```

static void OcrStatusUpdate(object sender, StatusUpdateEventArgs pageCompleteEventArgs)
{
    Console.WriteLine(new string('-', 50));
    Console.WriteLine("Page {0}", pageCompleteEventArgs.PageNumber);
    Console.WriteLine("Contains Text: {0}", pageCompleteEventArgs.TextAvailable);
    Console.WriteLine("Rotation: {0}", pageCompleteEventArgs.Rotation);
}
}
}

```

If you run the above code, you will get the following console output:



One notable difference from the above console output as compared to the one generated by the Aquaforest OCR engine is that the rotation is no longer displayed in 90° steps. Instead, you get the actual value of the rotation.

Now let's analyse the results of the OCR generated by the Extended OCR module:

Page 3 OCR results (with Extended module)

<p>Using the Remote Use the remote to control compatible media phone players. Answer/End Click the Call center button once. When answering calls speak in a normal manner. Answer/end calls on most phones by pressing the center button once. Please read your mobile device's user guide for more information on how to use the answer/end</p>	<p>Utilisation de la télécommande Utiliser la télécommande pour commander les lecteurs multimédia sur téléphones compatibles. Réponse Cliquer une fois / Fin sur le bouton d'appel central Parler d'une voix normale pour répondre aux coups de téléphone. Répondre à / terminer les appels sur la plupart des téléphones en appuyant une fois sur le bouton central. Prière de lire le guide d'utilisation</p>	<p>Uso del control remoto Usate il telecomando per agire sui vostro lettore multimediale compatibile. Invio/ Pulse el botón Fine central una vez chiamata Quando rispondete alle chiamate, parlate normalmente. I comandi Invio/Fine chiamate su lia maggior parte di telefoni vengono effettuati mediante pressione del pulsante centrale. Per ulteriori informazioni sull'uso del pulsante di invio/fine chiamata,</p>	<p>Verwendung der Fernsteuerung Die Fernsteuerung zur Bedienung kompatibler Medienwiedergabegeräte/ Handys verwenden. Anruf Mittlere annehmen/ beenden anklicken. Beim Telefonieren auf normale Weise sprechen. Bei den meisten Handys werden Anrufe angenommen/beendet, indem die mittlere Taste einmal gedrückt wird. In der Bedienungsanleitung Ihres Handys finden Sie</p>
--	---	--	---

<p>button, check for additional features or to troubleshoot usage problems. NOTE: On some phones it might be necessary to adjust the volume when switching from phone call to music. For more information about compatible models go to: www.shure.com</p>	<p>de l'appareil mobile pour trouver de plus amples renseignements sur l'emploi du bouton Réponse / Fin d'appel, pour connaître les fonctions supplémentaires ou pour dépanner les problèmes d'utilisation. REMARQUE : Sur certains téléphones il peut s'avérer nécessaire de régler le volume lorsqu'on passe des coups de téléphone à la musique. Pour de plus amples renseignements sur les modèles compatibles, visiter: www.shure.com.</p>	<p>su lia ricerca di funzioni aggiuntive o sull'individuazione di problemi d'uso, leggete la Guida u tente del vostro dispositivo mobile. NOTA - è possibile che su alcuni telefoni sia necessario regalare il volume quando si passa dalla telefonata all'ascolto di musica. Para más información sobre modelos compatibles, acuda a: www.shure.com</p>	<p>weitere Informationen über die Verwendung der Taste Annehmen/Beenden sowie über sonstige technische Eigenschaften oder die Störungssuche bei Nutzungsproblemen. HINWEIS: Bei manchen Telefonen ist es eventuell nötig, die Lautstärke anzupassen, wenn von einem Telefongespräch auf Musik umgeschaltet wird. Weitere Informationen über com compatible Modelle sind auf unserer Website zu finden: www.shure.com</p>
--	---	--	--

Page 6 OCR results (with Extended module)

Editeurs de logiciel indépendants

Les éditeurs de logiciels indépendants utilisent HATS Toolkit pour créer des applications personnalisées qui sont ensuite revendues à d'autres clients.

Accessibilité dans HATS

Les fonctions d'accessibilité permettent à un utilisateur souffrant d'un handicap physique tel qu'une mobilité réduite, un trouble de la vision, etc., d'utiliser les logiciels de manière satisfaisante. Etant donné qu'il constitue un ensemble de modules d'extension de Rational SOP, HATS bénéficie des fonctions d'accessibilité fournies par Rational SOP. Les principales fonctions d'accessibilité de Rational SOP sont les suivantes :

Rational SOP utilise les API Microsoft Active Accessibility (MSAA) pour rendre les éléments de l'interface utilisateur accessibles à la technologie dédiée à l'assistance.

Vous pouvez activer toutes les fonctions à partir du clavier au lieu d'utiliser la souris.

Remarque: Sur certains systèmes, il se peut que les traits de soulignement des touches de raccourci n'apparaissent pas dans la page des paramètres du composant Sous-fichier. Cette page est accessible à partir de Paramètres de projet > Rendu > Composants > Sous-fichier > Paramètres. Si c'est le cas sur votre système, pour afficher tous les traits de soulignement, utilisez les touches Alt+s pour accéder à la page, au lieu de cliquer sur le bouton Paramètres.

Vous pouvez utiliser un logiciel de lecteur d'écran tel que JAWS (Job Access With Speech) de Freedom Scientific et un synthétiseur de voix numérique pour reconnaître à l'ouïe ce qui s'affiche à l'écran.

Vous pouvez grossir l'affichage des vues graphiques.

Les polices ou couleurs définies par Rational SOP peuvent être configurées dans une boîte de dialogue à laquelle vous accédez en sélectionnant Fenêtre > Préférences > Informations générales > Présentation > Couleurs et polices.

The recognition results from the Extended module clearly show that the problem of OCRing a page with multiple languages has been overcome. Another advantage of using the Extended module is that the formatting of the resulting searchable document matches the text formatting in the source image more accurately than the Aquaforest module. Also, there is no need to make any changes to the properties.xml file in order to process documents or pages containing multiple languages.

7 Barcode Module

7.1 Overview

The barcode module supports decoding barcodes within images and PDF documents.

7.1.1 Supported Barcode Formats

The following barcode types are currently supported by the SDK:

- Aztec 2D barcode format.
- CODABAR 1D format.
- Code 39 1D format.
- Code 93 1D format.
- Code 128 1D format.
- Data Matrix 2D barcode format.
- EAN-8 1D format.
- EAN-13 1D format.
- ITF (Interleaved Two of Five) 1D format (Code 25).
- MaxiCode 2D barcode format.
- PDF417 format.
- QR Code 2D barcode format.
- RSS 14
- RSS EXPANDED
- UPC-A 1D format.
- UPC-E 1D format.
- UPC/EAN extension format.
- MSI
- Plessey

7.2 System Requirements

7.2.1 Supported Environments

- Windows Vista
- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2016

7.2.2 .NET Framework

.NET Version 4.5.2

7.3 Application Development and Deployment

7.3.1 Prerequisites

The table below shows the prerequisites needed for building applications using the Aquaforest OCR engine.

Application Platform	Minimum .NET Framework Version	Minimum Visual Studio Version
x86, x64 and Any CPU	.NET Framework 4.5.2	Visual Studio 2012

7.3.2 References

To use barcode decoding, a reference to `Aquaforest.BarcodeReader.Api.dll` and `Aquaforest.BarcodeReader.Common.dll` must be added to your application.

7.3.3 Deploying C# and VB.NET Applications

Ensure that the target system meets the System Requirements described in [section 7.2](#). Once, the target environment is set up, copy you solution/application files as well as the full contents of the SDK **bin** folder to the target environment. Set the `ResourceFolder` property of the `BarcodeReader` class inside your solution/application to the **bin** folder you copied to the target machine. For instance:

```
BarcodeReader reader = new BarcodeReader(settings);
reader.ResourceFolder = @"C:\TargetMachine\MyApp\bin";
```

The SDK contains an in-built functionality to detect if all the required assemblies and files required by the SDK are present. If they are not, an exception will be thrown listing all the files that are missing.

7.3.4 A Simple C# Example

```
using System;
using System.IO;
using System.Linq;
using Aquaforest.BarcodeReader.Api;
using Aquaforest.BarcodeReader.Common;

namespace GetOneBarcodePerPage
{
    class Program
    {
        static void Main(string[] args)
        {
            BarcodeReaderSettings settings = new BarcodeReaderSettings();
            settings.BarcodeFormats[BarcodeFormat.QR_CODE] = true;

            BarcodeReader reader = new BarcodeReader(settings);
            reader.ResourceFolder = Path.GetFullPath(@"..\..\..\..\bin\");

            reader.DecodeImage(Path.GetFullPath(@"..\..\..\..\documents\source\sample-
            barcodes.tif"));

            Console.WriteLine();
            Console.WriteLine("Decode Results");
            Console.WriteLine(new string('-', 30));

            foreach (var result in reader.DecodeResults.Where(r => r.Value.Success))
            {
                BarcodeResult r = result.Value.BarcodeResult;

                Console.WriteLine("Found barcode on page: {0}", result.Key);
                Console.WriteLine("Barcode Format: {0}", r.BarcodeFormat.ToString());
                Console.WriteLine("Text: {0}", r.Text);
                Console.WriteLine("Type: {0}", r.Type.ToString());
                Console.WriteLine();
            }
        }
    }
}
```

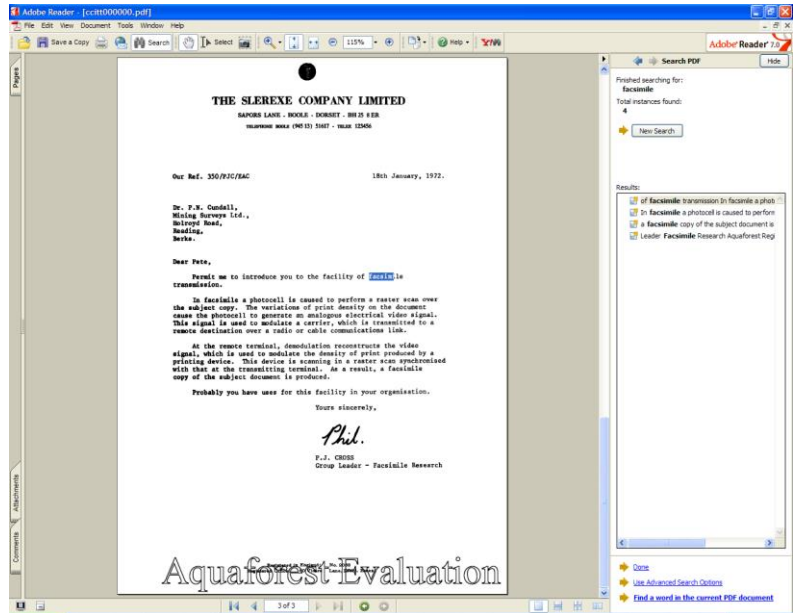
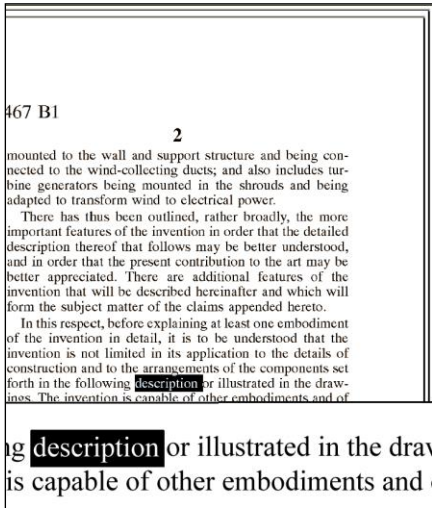
7.3.5 Barcode Module API

Refer to the **OCRSDK 2.2 (Aquaforest engine).chm** file found in the folder “[SDK installation path]\docs\help” to view all the properties and methods available for the Barcode module.

8 Background - Searchable PDFs

8.1 What is a Searchable PDF?

A searchable PDF file is a PDF file that includes text that can be searched upon using the standard Adobe Reader "search" functionality. In addition, the text can be selected and copied from the PDF. Generally, PDF files created from Microsoft Office Word and other documents are by their nature searchable as the source document contains text which is replicated in the PDF, but when creating a PDF from a scanned document and an OCR process needs to be applied to recognize the characters within the image.



8.2 Inside a Searchable PDF

In the context of Document Imaging, a searchable PDF will typically contain both the original scanned image plus a separate text layer produced from an OCR process. The text layer is defined in the PDF file as invisible, but can still be selected and searched upon. PDF files are able to store images using most of the native compression schemes used in TIFF files, so for example Group 4 TIFF files do not usually require any format conversion.

8.3 OCR Accuracy

A number of factors affect the accuracy of the text produced by the OCR process – 100% accuracy is certainly possible under good conditions but each of the following issues and OCR processing options will have an impact.

8.3.1 Original Image Quality

Although some pre-processing options such as despeckle and deskew can help in some cases, the visual quality of the original scan is of paramount importance.

8.3.2 Image DPI and Format

The image resolution should be at least 150 DPI for OCR processing, and preferably 300 DPI for optimal results, although for good quality scans 200 DPI is often sufficient. Non-lossy formats (TIFF Group 4, LZW etc.) are preferred over lossy formats such as JPEG.

8.3.3 Despeckle

This pre-processing option removes isolated "dots" within the image which can cause recognition problems, and makes the result image "cleaner".

8.3.4 Deskew

This option can improve OCR results by straightening crooked pages.

8.3.5 Auto-Rotate

OCR processing usually recognizes text written top-to-bottom, left-to-right, so pages that are orientated any other way (usually landscape pages) need to be re-oriented to enable recognition.

8.3.6 Graphics Areas

There are two options that can be used to control how the OCR engine processes parts of the document image that appear to be graphics areas.

To ensure that the OCR engine can be forced to process such areas there are two options:

"Treat all Graphics Areas as Text". This option will ensure the entire document is processed as text.

"Remove Box Lines in OCR Processing". This option is ideal for forms where sometimes boxes around text can cause an area to be identified as graphics. This option removes boxes from the temporary copy of the imaged used by the OCR engine. It does not remove boxes from the final image. Technically, this option removes connected elements with a minimum area (by default 100 pixels).

8.3.7 Language Settings

The language setting determines the set of characters that will be recognized, and the dictionary that will be used as a guide.

8.4 Hardware and Performance

8.4.1 CPU Power

The OCR process is highly CPU intensive and will benefit from being given as much CPU power as possible. As a guide about 2,000 pages per hour can be processed on a 3.0 GHz processor core, although this will vary according to the source document and OCR options chosen.

8.4.2 Exploiting Multiple CPUs

To take advantage of multiple cores, multiple OCR instances should be run in parallel.

8.4.3 Memory

Memory can be a limiting factor when creating the final PDF, in the case of very large documents. A rule of thumb would be to have 1GB – 1.5 GB of memory per processor core.

9 Acknowledgements

This product makes use of a number of Open Source components which are included in binary form. They are listed below.

Name	Homepage
BitMiracle.LibTiff.NET	Homepage GitHub
Cuneiform	n/a (Copyright (c) 1993-2008, Cognitive Technologies)
FreedImage.NET	Homepage
IKVM.NET	Homepage Sourceforge
Leptonica	Homepage
Libjpeg	Homepage
Libpng	Homepage
Libtiff	Homepage
log4Net	Homepage
PDFBox	Homepage
veraPDF	Homepage GitHub
Zlib	Homepage
ZXing.NET	Homepage