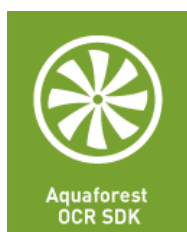

Aquaforest OCR SDK for .NET Reference Guide



Version 2.10
April 2016

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.....	3
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
3.1.3	Methods	11
3.2	OCR CLASS.....	13
3.2.1	Constructor.....	13
3.2.2	Properties	13
3.2.3	Methods	18
3.2.4	Events	20
3.2.5	Subscribing to StatusUpdate using C#.....	20
3.2.6	Subscribing to StatusUpdate using VB.NET	20
3.2.7	Enumerations.....	21
3.3	STATUSUPDATEEVENTARGS CLASS	22
3.3.1	Constructor.....	22
3.3.2	Properties	22
3.3.3	Words Class.....	23
3.3.4	Constructor.....	23
3.3.5	Properties	23
3.3.6	Methods	23
3.3.7	WordData Class	23
3.3.8	Properties	23

3.4	ERROR HANDLING.....	24
3.5	PDFMERGER CLASS	24
3.5.1	Constructor.....	24
3.5.2	Methods.....	24
3.6	DISPOSAL AND TEMPORARY FILES FOLDERS	24
3.7	MULTI-THREADED APPLICATIONS	24
3.8	ADVANCED PRE-PROCESSING.....	24
3.9	PROPERTIES FILE.....	27
4	AQUAFOREST EXTENDED OCR MODULE	30
4.1	OVERVIEW.....	30
4.2	SYSTEM REQUIREMENTS	30
4.2.1	Supported Environments.....	30
4.2.2	.NET Framework.....	30
4.2.3	Visual C++ Runtime.....	30
4.3	FOLDERS.....	30
4.4	APPLICATION DEVELOPMENT AND DEPLOYMENT.....	31
4.4.1	Prerequisites.....	31
4.4.2	References.....	31
4.4.3	Properties.xml	31
4.4.4	Deploying C# and VB.NET Applications	32
5	AQUAFOREST EXTENDED OCR MODULE API REFERENCE	33
5.1	PREPROCESSOR CLASS	33
5.1.1	Constructor.....	33
5.1.2	Properties	33
5.2	OCR CLASS.....	37
5.2.1	Constructor.....	37
5.2.2	Properties	38
5.2.3	Methods.....	43
5.2.4	Events	47
5.2.5	Subscribing to StatusUpdate	47
5.3	STATUSUPDATEEVENTARGS CLASS	48
5.3.1	Constructor.....	48
5.3.2	Properties	48
5.3.3	Words Class.....	48
5.3.4	Words Constructor	48
5.3.5	Words Properties	48
5.3.6	Words Methods	49
5.3.7	WordData Class	49
5.3.8	WordData Properties.....	49
5.3.9	CharacterData Class	49
5.3.10	CharacterData Properties	49
5.4	ENUMERATIONS.....	50
6	AQUAFOREST OCR VS. EXTENDED OCR	58
6.1	DIFFERENCES BETWEEN AQUAFOREST OCR AND EXTENDED OCR	58
6.1.1	References.....	58
6.1.2	Ocr Methods	58
6.1.3	Ocr Properties.....	58
6.1.4	PreProcessor Methods.....	59
6.1.5	PreProcessor Properties.....	59

6.2	CREATING A SIMPLE APPLICATION	60
6.2.1	Using Aquaforest SDK and Visual Studio 2012	61
6.2.2	Converting to Extended OCR	71
7	BARCODE MODULE	76
7.1	OVERVIEW	76
7.1.1	Supported Barcode Formats	76
7.2	SYSTEM REQUIREMENTS	76
7.2.1	Supported Environments	76
7.2.2	.NET Framework	76
7.3	APPLICATION DEVELOPMENT AND DEPLOYMENT	76
7.3.1	Prerequisites	76
7.3.2	References	77
7.3.3	Deploying C# and VB.NET Applications	77
7.3.4	A Simple C# Example	77
8	BARCODE MODULE API REFERENCE	78
8.1	BARCODEREADER CLASS	78
8.1.1	Constructor	78
8.1.2	Properties	78
8.1.3	Methods	79
8.2	BARCODEREADERSETTINGS CLASS	80
8.2.1	Constructor	80
8.2.2	Properties	80
8.3	BARCODERESULT CLASS	82
8.3.1	Properties	82
8.4	BARCODERESULTPOINT CLASS	83
8.4.1	Properties	83
8.5	DECODERESULT CLASS	83
8.5.1	Properties	83
8.6	ENUMERATIONS	84
9	BACKGROUND - SEARCHABLE PDFS	87
9.1	WHAT IS A SEARCHABLE PDF?	87
9.2	INSIDE A SEARCHABLE PDF	87
9.3	OCR ACCURACY	87
9.3.1	Original Image Quality	87
9.3.2	Image DPI and Format	87
9.3.3	Despeckle	88
9.3.4	Deskew	88
9.3.5	Auto-Rotate	88
9.3.6	Graphics Areas	88
9.3.7	Language Settings	88
9.4	HARDWARE AND PERFORMANCE	88
9.4.1	CPU Power	88
9.4.2	Exploiting Multiple CPUs	88
9.4.3	Memory	88
10	ACKNOWLEDGEMENTS	89

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 co-ordinates.
- 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.
- Apply stamps to output PDF files.
- 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
Stamps on PDF Output		✓	✓	✓
Unlimited Threads			✓	✓
Advanced MRC Compressed PDF Output			✓	✓
Advanced Pre-Processing			✓	✓
Support for 127 languages				✓
Asian language support				✓
Support for multiple languages within a single document from the same character set				✓
Multiple document output formats: CSV, DOCX, EPUB, EXCELML, HTML, OPEN DOCUMENT TEXT, PDF, RTF, TXT, WORDML, XLSX and XPS				✓
Multiple PDF version output support				✓
Intelligent High Quality Compression				✓

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

2.2.2 .NET Framework

.NET Version 3.5

2.2.3 Visual C++ Runtime

The Visual C++ 2012 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 ++ 2012 x86	.NET Framework 3.5	Visual Studio 2008
x64	VC ++ 2012 x86 and VC++ 2012 x64	.NET Framework 3.5	Visual Studio 2008
Any CPU	VC ++ 2012 x86 and VC++ 2012 x64	.NET Framework 3.5	Visual Studio 2008

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.

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++ 2012 Redistributable package and .NET Version 3.5 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 2008 and conversion to Visual Studio 2010 is handled automatically by that IDE. Conversion to Visual Studio 2005 is also possible. In this case you must ensure that version 3.5 of the .NET Framework is installed on your system and then simply copy the classes/forms into a new project and add the missing references.

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

Property	Description
bool Autorotate	Auto-rotate the image – this will ensure all text oriented normally. The default value is false (disabled). Note: When using a PDF source Autorotation will be disabled on any pages already containing text.
int Binarize	This value should generally only be used under guidance from technical support. It can control the way that color images are processed and force binarization with a particular threshold. A value of 200 has been shown to generally give good results in testing, but this should be confirmed with "typical" customer documents. By setting this to -1 an alternative method is used which will attempt to separate the text from any background images or colors. This can give improved OCR results for certain documents such as newspaper and magazine pages.
int BlackPixelLimit	Contact technical support (support@aquaforest.com) for guidance on using this property.
int BlankPageThreshold	Use this to set the minimum number of "On Pixels" that must be present in the image for a page not to be considered blank. A value of -1 will turn off blank page detection. A value of 100 produced reasonable blank page detection in testing, but the validity of this should be confirmed using "typical" source documents.
int BoxSize	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 (in pixels and defined by this property). This option is currently only applied for bitonal images.

Property	Description
int Despeckle	Despeckle the image – The method removes all disconnected elements within the image that have height or width in pixels less than the specified figure. The maximum value is 9 and the default value is 0.
bool Deskew	Deskew (straighten) the image. The default value is false (disabled).
int GrayScaleQuality	This option should generally only be used under guidance from technical support.
string 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
bool LibTiffSavePageAsBmp	Sometimes if there is an image which is 1bpp and has LZW compression, the pre-processing can cause the colour of the image to be inverted (black to white and white to black). Set this to true to avoid this.
float MaxDeskew	Maximum angle by which a page will be de-skewed. This option should generally only be used under guidance from technical support.
float MinDeskewConfidence	This option should generally only be used under guidance from technical support.
string Morph	Image Morphology. This option should generally only be used under guidance from technical support.
bool MRC	This enables Mixed Raster Compression which can dramatically reduce the output size of PDFs comprising color scans. Note that this option is only suitable when the source is not a PDF.
int MRCBackgroundFactor	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
int MRCForegroundFactor	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
int MRCQuality	JPEG quality setting (percentage value 1 - 100) for use in saving the background and foreground images. Default value is 75

Property	Description
bool 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.
PdfToImageBpp PdfToImageBpp	The Bits Per Pixel to use for the rasterized PDF page when using engine 1. This only applies for documents that are not processed using Native mode. The default value for this property is taken from the PDF page. <i>Note: This should only be set with guidance from technical support.</i>
PdfToImageCompression PdfToImageCompression	The DPI to set to the images extracted or rasterized from each page of the source PDF file. These images are then OCR'd to create the searchable PDF. The default value for this is taken from each page in the source PDF file.
PdfToImageDpi PdfToImageDpi	The DPI to set to the images rasterized from each page of the source PDF file. These images are then OCR'd to create the searchable PDF. The default value for this is taken from each page in the source PDF file. <i>Note: This property is used in PdfToImageEngine 1 only.</i>
int PdfToImageEngine	The engine to use for extracting or generating images from the source PDF file to use for OCR'ing. Valid values are 0 and 1.
int PdfToImageMinRes	The minimum resolution of the rasterized images. If the resolution retrieved from the PDF page is lower than this value, it will be set to this value. The default value for this is property 150. <i>Note: This property is used in PdfToImageEngine 1 only.</i>
int PdfToImageMaxRes	The maximum resolution of the rasterized images. If the resolution retrieved from the PDF page is bigger than this value, it will be set to this value. The default value for this property is 600. <i>Note: This property is used in PdfToImageEngine 1 only.</i>
int 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. <i>Note: This property is used in PdfToImageEngine 1 only.</i>

Property	Description
bool RemoveLines	When set to true this will enable the removal of lines. This feature is particularly useful where pages contain tables and underlining which can prevent the OCR engine from recognising characters. The lines are removed only from the image used in OCR and not from the image used in the final PDF if PDF creation is enabled.
bool SavePreDespeckle	This will use the original image (i.e. before applying pre-processing) in the output pdf. The default value is true.
bool Tables	This option when set to true, tries to OCR within table cells
int TextLayerFilterHeight	Contact technical support (support@aquaforest.com) for guidance on using this property.
int TextLayerFilterHeightInverted	Contact technical support (support@aquaforest.com) for guidance on using this property.
float TextLayerFilterPercentage	Contact technical support (support@aquaforest.com) for guidance on using this property.
float TextLayerFilterPercentageInverted	Contact technical support (support@aquaforest.com) for guidance on using this property.
float TextLayerFilterRatio	Contact technical support (support@aquaforest.com) for guidance on using this property.
float TextLayerFilterRatioInverted	Contact technical support (support@aquaforest.com) for guidance on using this property.
int TextLayerFilterWidth	Contact technical support (support@aquaforest.com) for guidance on using this property.
int TextLayerFilterWidthInverted	Contact technical support (support@aquaforest.com) for guidance on using this property.
int TextLayerMaxBoxes	Contact technical support (support@aquaforest.com) for guidance on using this property.

3.1.3 Methods

Method	Description
ConfigurePDFStamp(string prefix, string suffix, Nullable<int> start, Nullable<int> digits, PagePositionEnum position, StampType stampType)	Using this method stamps can be configured to be added to each page of the PDF output. The stamps contain one or more of the following: Prefix – a string to be added to the beginning of the stamp, before the number section. Start – the value that the number portion of the stamp should start at. The number portion will be incremented by 1 each page. Digits – a value indicating the minimum length that the number portion of the stamp should be displayed as. Preceding 0's will be used to pad any numbers less than this whilst numbers greater than this will be displayed in full. Suffix - a string to be added to the end of the stamp, after the number section. Thus a stamp with Prefix = "Beginning", Start = "1", Digits = "4" and Suffix = "End" would produce the text "Beginning0001End" on the first page. Any

	one of these can be set to null resulting in the exclusion of that part from the final text. Additionally the stamp can be added either as visible searchable text or as an image and can be positioned in one of the following: • Top Left • Top Centre • Top Right • Centre Left • Centre • Centre Right • Bottom Left • Bottom Centre • Bottom Right
--	---

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

Property	Description
bool AdvancedPreProcessing	When set to true, this will enable the advanced pre-processing functionality. See section 3.8 for details.
int AquaforestImageTimeout	Contact technical support (support@aquaforest.com) for guidance on using this property.
bool Author	Set a custom Author in the output PDF document properties.
bool ConvertToTiff	Each page in the PDF document is rasterized to a TIFF image
bool CreateProcess	Set this to true if you want to launch process through pinvoke.
DateTime CreationDate	Set a custom creation date in the output PDF document properties. Note: If you are setting the date through the properties.xml, then the date string should be in the format 'yyyy-MM-dd HH:mm:ss'.
int CurrentPage	Returns the current page for which the OCR has been performed. This is useful only when using Recognize() in another thread.
bool DeleteTemporaryFilesOnPageCompletion	When set to true the temporary files generated for each page during OCR processing will be removed as soon as the OCR engine has finished with them. Note: The OCR engine is finished with the temporary files for a page as soon as the output for that page is added to the overall output. If you wish to use functionality such as ReadPageWords, GetPageImage, etc. then this will require that the temporary files are available for the page requested and so will fail if DeleteTemporaryFilesOnPageCompletion is true.
int DictionaryLookup	Contact technical support (support@aquaforest.com) for guidance on using this property.
bool 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.

Property	Description
bool EnableConsoleOutput	If enabled then progress messages will be sent to the console. Default is false.
int EnableDebugOutput	If set to a value greater than 0 (default value) debug messages will be written to the console output. Please contact Aquaforest for guidance on suitable values if you need to generate debug output.
bool EnablePDFOutput	Enables or disables the production of Portable Document Format output. Default value is false (disabled).
bool EnableRTFOutput	Enables or disables the production of Rich Text Format output. Default value is false (disabled).
bool EnableTextOutput	Enables or disables the production of simple text final output. Default value is true (enabled).
int EndPage	Sets the last page of the source file that the OCR process will be run to (for a multipage source). Throws an ArgumentOutOfRangeException if a source file has not been set already (by using the ReadBMPSource or ReadTIFFSource method prior to setting this property) or if the page is greater than the number of pages in the source. By default the whole of the document will be processed.
int ErrorMode	Contact technical support (support@aquaforest.com) for guidance on using this property.
int FlipDetect	Contact technical support (support@aquaforest.com) for guidance on using this property.
bool HandleExceptionsInternally	When set to true the Ocr object will catch any exceptions for method calls and simply return false from the method. The exceptions caught are stored in the LastException property overwriting any previous value.
int Heuristics	Contact technical support (support@aquaforest.com) for guidance on using this property.
SupportedLanguages Language	Sets the language to be used for the OCR processing. This takes a value from the enumeration SupportedLanguages which is defined in the API. Default language is English.
Exception LastException	Stores last exception caught by the Ocr object.
string License	Sets the license key.

Property	Description														
LicenseInfo	Get the current license information. <i>Note:</i> This needs to be called after setting <code>Ocr.License</code>. <table> <tr> <th>Property</th><th>Description</th></tr> <tr> <td>bool AsianOCREnabled</td><td>Get whether Asian OCR is enabled.</td></tr> <tr> <td>bool IHQCEnabled</td><td>Get whether Intelligent High Quality Compression (IHQC) is enabled.</td></tr> <tr> <td>bool IsTrial</td><td>Get whether the license is trial</td></tr> <tr> <td>string LicenseType</td><td>Get the <u>license type</u>: Basic, Standard, Advanced or Extended.</td></tr> <tr> <td>int LicenseType</td><td>Get the maximum number of threads allowed to run concurrently</td></tr> <tr> <td>string SDKVersion</td><td>Get the version of the SDK that this license is for.</td></tr> </table>	Property	Description	bool AsianOCREnabled	Get whether Asian OCR is enabled.	bool IHQCEnabled	Get whether Intelligent High Quality Compression (IHQC) is enabled.	bool IsTrial	Get whether the license is trial	string LicenseType	Get the <u>license type</u> : Basic, Standard, Advanced or Extended.	int LicenseType	Get the maximum number of threads allowed to run concurrently	string SDKVersion	Get the version of the SDK that this license is for.
Property	Description														
bool AsianOCREnabled	Get whether Asian OCR is enabled.														
bool IHQCEnabled	Get whether Intelligent High Quality Compression (IHQC) is enabled.														
bool IsTrial	Get whether the license is trial														
string LicenseType	Get the <u>license type</u> : Basic, Standard, Advanced or Extended.														
int LicenseType	Get the maximum number of threads allowed to run concurrently														
string SDKVersion	Get the version of the SDK that this license is for.														
string LogFilePath	Gets the path of the log file where output is written to if <code>DebugOutput</code> is set to <code>DebugLevels.LOG_TO_FILE</code>. <i>Note:</i> The value for this property must be accessed after calling '<code>Ocr.Recognize</code>'.														
DateTime ModifiedDate	Set a custom modified date in the output PDF document properties. <i>Note:</i> If you are setting the date in through the <code>properties.xml</code>, then the date string should be in the format '<code>yyyy-MM-dd HH:mm:ss</code>'.														
int MrcTimeout	The maximum time (in milliseconds) to compress the image in a page.														
int NumberOfPages	Returns the number pages in a document.														
int OcrProcessSetupTimeout	The maximum time (in milliseconds) to setup the OCR process.														
int OcrTimeout	The maximum time (in milliseconds) to OCR a page.														
bool 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.														
bool OptimiseOcr	[Deprecated] Use <code>AdvancedPreProcessing</code> instead.														

Property	Description
int PipeClientConnectionTimeout	Contact technical support (support@aquaforest.com) for guidance on using this property.
string PropertiesPath	The path of the properties.xml file. This is an optional property that should only be changed if you have specific needs to keep the properties.xml file in another location other than the resource folder. Note: This property should be set before setting the ResourceFolder property.
bool ReadBmpViaFreelImage	Contact technical support (support@aquaforest.com) for guidance on using this property.
bool RemoveExistingPDFText	If set to true, it will result in the removal of any existing text from the output PDF. Note: when PDF output is generated from a PDF source it is a copy of the PDF that is manipulated rather than generating a new one. This approach offers several advantages such as potential size savings and performance enhancements.
string ResourceFolder	This property must be set to the location containing the resources of the OCR SDK (usually 'C:\Aquaforest\OCRSDK\bin').
int RestartEngineEvery	Contact technical support (support@aquaforest.com) for guidance on using this property.
bool RetainBookmarks	Retains any bookmarks from the source file in the output when using ConvertToTiff.
bool RetainCreationDate	Retains the creation date of the source file in the output PDF document properties.
bool RetainMetadata	Retains any metadata from the source file in the output when using ConvertToTiff.
bool RetainModifiedDate	Retains the modified date of the source file in the output PDF document properties.
int StartPage	Sets the first page of the source file that the OCR process will be begin from (for a multipage source). Throws an ArgumentOutOfRangeException if a source file has not been set already (by using the ReadBMPSource or ReadTIFFSource method prior to setting this property) or if the page is greater than the number of pages in the source. By default the whole of the document will be processed.
string TempFolder	Specifies a temporary folder for storing bitmap images and intermediate output during OCR processing. If this is not specified, the first of the following environment variables that is defined will be used: TMP, TMPDIR, TEMP.
int TidyUpMode	Contact technical support (support@aquaforest.com) for guidance on using this property.

Property	Description
bool UseAquaforestImagingFontSizing	Contact technical support (support@aquaforest.com) for guidance on using this property. Default value is false.
string Version	Returns the version of the SDK being used.
float WordMatchThreshold	Contact technical support (support@aquaforest.com) for guidance on using this property.

3.2.3 Methods

Method	Description
void Abort()	Stops processing of an ongoing call to Recognize. Processing will stop on completion of any ongoing page.
void ConvertPDFToPDFA(string source, string target, PDFAVersion version)	Converts a PDF file to PDF/A without going through the OCR process.
void DeleteTemporaryFiles()	Removes temporary files created during the OCR processing from the system. Note, do not call this before you have completely finished processing a file.
Image GetPageImage(int pageNumber)	Returns a System.Drawing.Image containing the processed image.
void Recognize(PreProcessor preProcessor)	Performs any pre-processing defined in the PreProcessor object and then carries out OCR processing on the pre-processed image.
bool ReadBMPSource(string fileName)	Checks for the existence of the source file and sets up the OCR engine for handling the bitmap image.
bool ReadBMPSource(Stream source)	Read a bitmap source from a System.IO.Stream source for OCR.
bool ReadImageSource(Image image)	Reads an Image object checking the number of frames (pages).
bool ReadPDFSource(string fileName)	Checks for the existence of the source file and sets up the OCR engine for handling the PDF.
bool ReadPDFSource(Stream source)	Read a PDF source from a System.IO.Stream source for OCR.
bool ReadPDFSource(string filename, string password)	Checks for the existence of the source file and sets up the OCR engine for handling the secure PDF for which the password is provided. If PDF output is generated from this the output will have no security settings defined.
bool ReadPDFSource(Stream source, string password)	Read a password protected PDF source from a System.IO.Stream source for OCR.
string ReadDocumentString()	Returns a string containing the words from all pages processed.
string ReadPageString(int pagenumber)	Returns a string containing the words from the specified page.
string ReadPageString(int pagenumber, Rectangle region)	Returns a string containing the words for the specified page where the words are fully enclosed in the bounds of the region specified.
Words ReadPageWords(int pagenumber)	Returns an instance of the Words class for the specified page.
Words ReadPageWords(int pagenumber, Rectangle region)	Returns an instance of the Words class for the specified page where the words are fully enclosed in the bounds of the region specified.
void ReadTIFFSource(string fileName)	Checks for the existence of the source file and sets up the OCR engine for handling the TIFF image.
bool ReadTIFFSource(Stream source)	Read a TIFF source from a System.IO.Stream source for OCR.
bool SavePDFAOutput(string fileName, bool overwriteExisting, PDFAVersion version)	Saves PDF/A file with the name provided.
bool SavePDFOutput(string fileName, bool overwriteExisting)	Saves the output to a PDF file with the name specified. If any text was extracted then this will be searchable in the PDF.

Method	Description
<code>bool SaveRTFOutput(string fileName, bool overwriteExisting)</code>	Saves the output to a RTF file with the name provided.
<code>bool SaveTextOutput(string fileName, bool overwriteExisting)</code>	Saves the text extracted to a simple text file with the name provided.

3.2.4 Events

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

3.2.5 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;`

3.2.6 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
```


3.2.7 Enumerations

Enumeration	Description																																																			
DebugLevels	Using the default settings various exceptions can be thrown by the Ocr object so these should be trapped within the calling code. Below are the different DebugLevels that can be set: <table><tr><th>Member name</th><th>Value</th><th>Description</th></tr><tr><td>NONE</td><td>0</td><td></td></tr><tr><td>AUTOROTATE_INFO</td><td>1</td><td>Autorotate information</td></tr><tr><td>AUTOROTATE_WORDS</td><td>2</td><td>Words used in autorotate decision</td></tr><tr><td>DEAD_PROCESS</td><td>4</td><td>Notification of Dead Child Process</td></tr><tr><td>ERRORS</td><td>8</td><td>Handled errors</td></tr><tr><td>EXCEPTIONS</td><td>16</td><td>Stack Trace and exception messages</td></tr><tr><td>IMAGE_PREPROCESSING</td><td>32</td><td>Image pre-processing</td></tr><tr><td>LEAVE_TEMP_FILES_IN_PLACE</td><td>64</td><td>Do not delete temporary files</td></tr><tr><td>MESSAGES</td><td>128</td><td>Interprocess messages</td></tr><tr><td>PDF_IMAGE_EXTRACTION</td><td>256</td><td>PDF image extraction</td></tr><tr><td>PIPE_EVENTS</td><td>512</td><td>Pipe connections/disconnections events</td></tr><tr><td>PROGRESS_UPDATES</td><td>1024</td><td>General progress messages</td></tr><tr><td>TEMP_FILE_OPERATION</td><td>2048</td><td>Temp File Operations</td></tr><tr><td>TIMEOUTS</td><td>4096</td><td>Timeout messages</td></tr><tr><td>LOG_TO_FILE</td><td>8192</td><td>Log output to file</td></tr><tr><td>STOP_ON_ERROR</td><td>16384</td><td>Exit application on certain errors</td></tr></table>	Member name	Value	Description	NONE	0		AUTOROTATE_INFO	1	Autorotate information	AUTOROTATE_WORDS	2	Words used in autorotate decision	DEAD_PROCESS	4	Notification of Dead Child Process	ERRORS	8	Handled errors	EXCEPTIONS	16	Stack Trace and exception messages	IMAGE_PREPROCESSING	32	Image pre-processing	LEAVE_TEMP_FILES_IN_PLACE	64	Do not delete temporary files	MESSAGES	128	Interprocess messages	PDF_IMAGE_EXTRACTION	256	PDF image extraction	PIPE_EVENTS	512	Pipe connections/disconnections events	PROGRESS_UPDATES	1024	General progress messages	TEMP_FILE_OPERATION	2048	Temp File Operations	TIMEOUTS	4096	Timeout messages	LOG_TO_FILE	8192	Log output to file	STOP_ON_ERROR	16384	Exit application on certain errors
Member name	Value	Description																																																		
NONE	0																																																			
AUTOROTATE_INFO	1	Autorotate information																																																		
AUTOROTATE_WORDS	2	Words used in autorotate decision																																																		
DEAD_PROCESS	4	Notification of Dead Child Process																																																		
ERRORS	8	Handled errors																																																		
EXCEPTIONS	16	Stack Trace and exception messages																																																		
IMAGE_PREPROCESSING	32	Image pre-processing																																																		
LEAVE_TEMP_FILES_IN_PLACE	64	Do not delete temporary files																																																		
MESSAGES	128	Interprocess messages																																																		
PDF_IMAGE_EXTRACTION	256	PDF image extraction																																																		
PIPE_EVENTS	512	Pipe connections/disconnections events																																																		
PROGRESS_UPDATES	1024	General progress messages																																																		
TEMP_FILE_OPERATION	2048	Temp File Operations																																																		
TIMEOUTS	4096	Timeout messages																																																		
LOG_TO_FILE	8192	Log output to file																																																		
STOP_ON_ERROR	16384	Exit application on certain errors																																																		
PDFAVersion	The PDFA versions available in the SDK <table><tr><th>Member name</th><th>Value</th><th>Description</th></tr><tr><td>PDF_A1b</td><td>1</td><td>PDF/A-1b</td></tr><tr><td>PDF_A2b</td><td>2</td><td>PDF/A-2b</td></tr><tr><td>PDF_A3b</td><td>3</td><td>PDF/A-3b</td></tr></table>	Member name	Value	Description	PDF_A1b	1	PDF/A-1b	PDF_A2b	2	PDF/A-2b	PDF_A3b	3	PDF/A-3b																																							
Member name	Value	Description																																																		
PDF_A1b	1	PDF/A-1b																																																		
PDF_A2b	2	PDF/A-2b																																																		
PDF_A3b	3	PDF/A-3b																																																		
SupportedLanguages	This enumeration includes all of the languages currently supported by the API. <table><tr><th>Member name</th><th>Value</th></tr><tr><td>English</td><td>0</td></tr><tr><td>German</td><td>1</td></tr><tr><td>French</td><td>2</td></tr><tr><td>Russian</td><td>3</td></tr></table>	Member name	Value	English	0	German	1	French	2	Russian	3																																									
Member name	Value																																																			
English	0																																																			
German	1																																																			
French	2																																																			
Russian	3																																																			

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

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.

Property	Description
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
void Append(string pdfFileToAdd)	Appends the document specified to the in memory PDF document.
void Close()	Writes the output to the file specified in the constructor.
void Dispose()	Clears any resources not yet released. This is useful if Close (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 of 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.

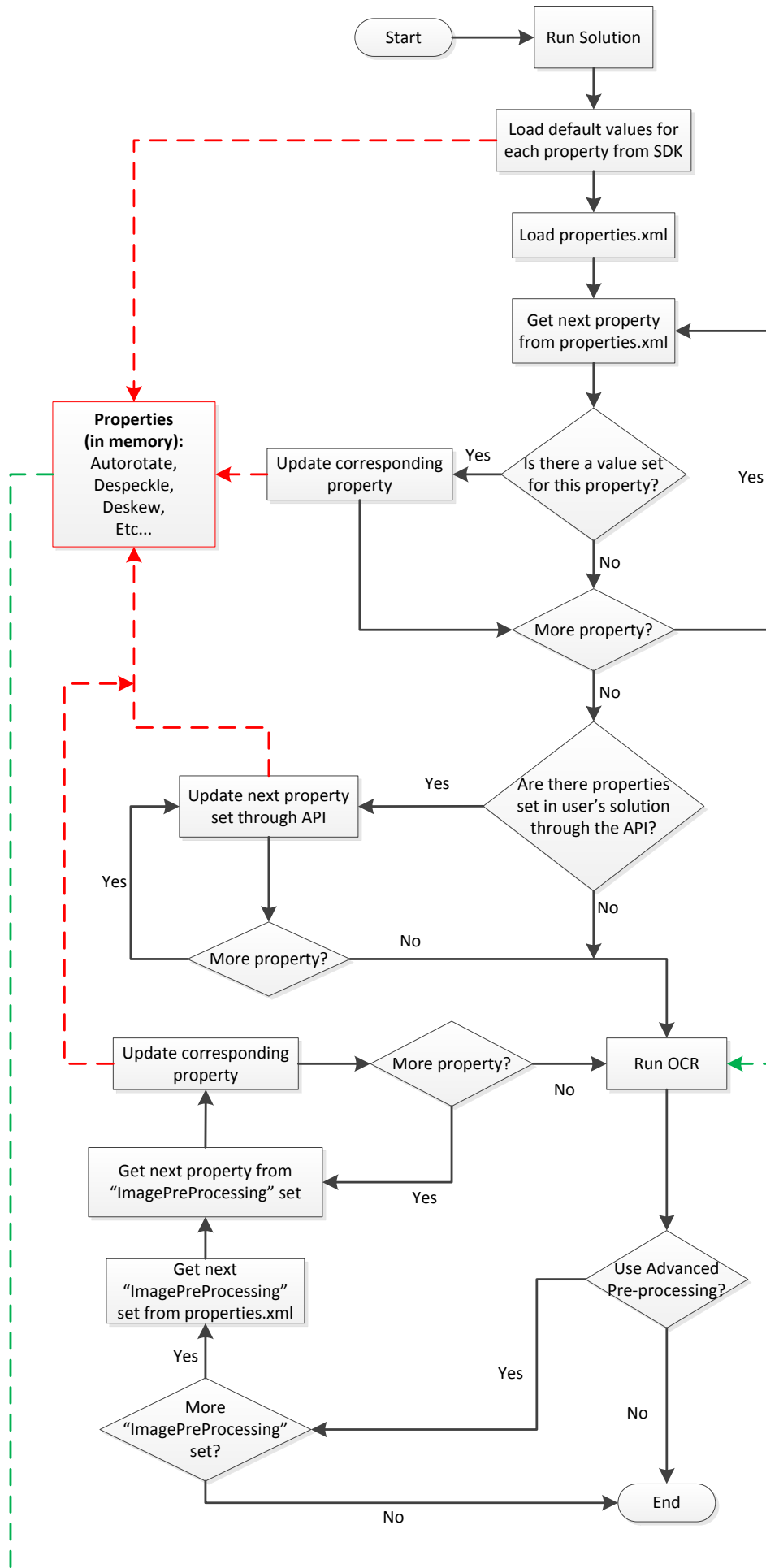
3.8 Advanced pre-processing

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 vales 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.

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 127 languages. See [section 5.4](#) for more details.
- Optional Asian language support: Standard Chinese, Traditional Chinese, Korean and Japanese.
- 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

4.2.2 .NET Framework

.NET Version 4

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	Visual Studio 2010
x64	VC++ 2010 x64	.NET Framework 4	Visual Studio 2010
Any CPU	VC ++ 2010 x86 and VC++ 2010 x64	.NET Framework 4	Visual Studio 2010

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.

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.

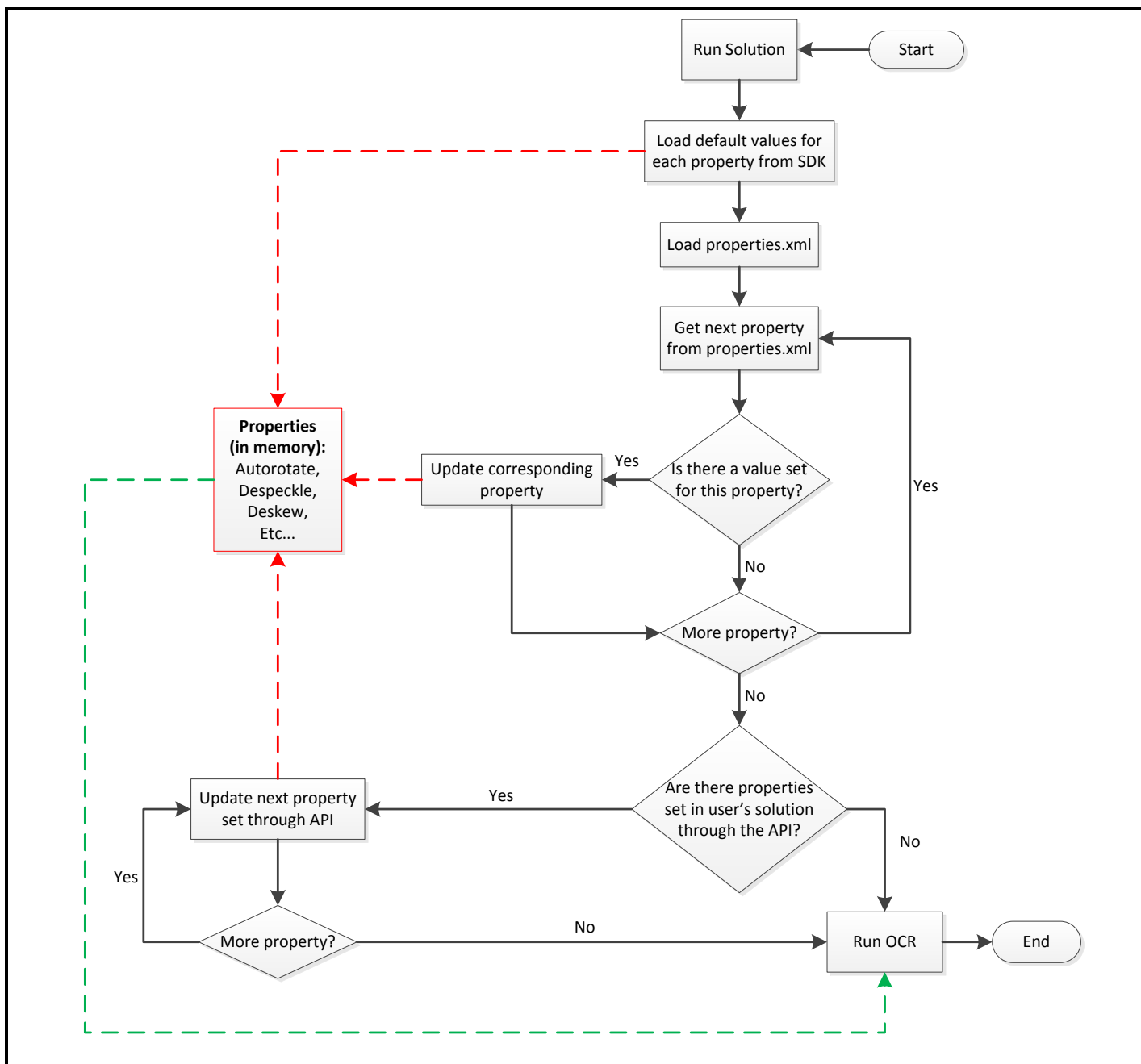
4.4.3 Properties.xml

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.

A list of the pre-processing settings available is described in section [5.1.2](#) below.

5.1.1 Constructor

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

5.1.2 Properties

Setting	Description					
AdvancedDespeckle	The AdvancedDespeckle class provides advanced image noise reduction features by image de-speckle filter.					
	<table><tr><th>Property</th><th>Description</th></tr><tr><td>bool RemoveWhitePixels</td><td>By default, 'AdvancedDespeckle' removes black pixels. If this setting is set to 'true', white pixels will be removed instead of black pixels.</td></tr><tr><td>int SpeckleSize</td><td>The size of speckles to remove.</td></tr></table>	Property	Description	bool RemoveWhitePixels	By default, 'AdvancedDespeckle' removes black pixels. If this setting is set to 'true', white pixels will be removed instead of black pixels.	int SpeckleSize
Property	Description					
bool RemoveWhitePixels	By default, 'AdvancedDespeckle' removes black pixels. If this setting is set to 'true', white pixels will be removed instead of black pixels.					
int SpeckleSize	The size of speckles to remove.					
bool Autorotate	When set to 'true', it will automatically rotate pages so that text flows from left to right. The default value is 'false'.					

Setting	Description												
Binarization	The Binarization class offers image binarization features. Binarization is the first stage in image recognition. It converts the input image into a black and white image for faster processing. <table> <tr> <th>Property</th><th>Description</th></tr> <tr> <td>bool Binarize</td><td>Whether or not to perform binarization on the document. The default value is 'false'.</td></tr> <tr> <td>int Brightness</td><td> The brightness threshold (higher values will darker the result). The default value is 0. <i>Note:</i> For this setting to work, Binarize needs to be set to 'true'. </td></tr> <tr> <td>int Contrast</td><td> The contrast threshold (lower values will darker the result). The default value is 40. <i>Note:</i> For this setting to work, Binarize needs to be set to 'true'. </td></tr> <tr> <td>int SmoothingLevel</td><td>Smoothing may be useful to binarize text with a colored background in order to avoid noisy pixels. (0 disables smoothing, higher values smoothes more).</td></tr> <tr> <td>int Threshold</td><td>Sets the threshold for fixed threshold binarization (0 for automatic threshold computation). The default value is 128.</td></tr> </table>	Property	Description	bool Binarize	Whether or not to perform binarization on the document. The default value is 'false'.	int Brightness	The brightness threshold (higher values will darker the result). The default value is 0. <i>Note:</i> For this setting to work, Binarize needs to be set to 'true'.	int Contrast	The contrast threshold (lower values will darker the result). The default value is 40. <i>Note:</i> For this setting to work, Binarize needs to be set to 'true'.	int SmoothingLevel	Smoothing may be useful to binarize text with a colored background in order to avoid noisy pixels. (0 disables smoothing, higher values smoothes more).	int Threshold	Sets the threshold for fixed threshold binarization (0 for automatic threshold computation). The default value is 128.
Property	Description												
bool Binarize	Whether or not to perform binarization on the document. The default value is 'false'.												
int Brightness	The brightness threshold (higher values will darker the result). The default value is 0. <i>Note:</i> For this setting to work, Binarize needs to be set to 'true'.												
int Contrast	The contrast threshold (lower values will darker the result). The default value is 40. <i>Note:</i> For this setting to work, Binarize needs to be set to 'true'.												
int SmoothingLevel	Smoothing may be useful to binarize text with a colored background in order to avoid noisy pixels. (0 disables smoothing, higher values smoothes more).												
int Threshold	Sets the threshold for fixed threshold binarization (0 for automatic threshold computation). The default value is 128.												
BlankPageRemoval	This class offers blank page detection and removal features. <table> <tr> <th>Property</th><th>Description</th></tr> <tr> <td>bool DetectBlankPage</td><td>Detect blank pages in the source document. The default is 'true'.</td></tr> <tr> <td>bool RemoveBlankPage</td><td>Remove blank pages from the output document. The default value is 'false'.</td></tr> <tr> <td>int Sensitivity</td><td>The sensitivity (1 -100) for blank page detection. With high sensitivity, less blank pages are detected. Default value is 1.</td></tr> </table>	Property	Description	bool DetectBlankPage	Detect blank pages in the source document. The default is 'true'.	bool RemoveBlankPage	Remove blank pages from the output document. The default value is 'false'.	int Sensitivity	The sensitivity (1 -100) for blank page detection. With high sensitivity, less blank pages are detected. Default value is 1.				
Property	Description												
bool DetectBlankPage	Detect blank pages in the source document. The default is 'true'.												
bool RemoveBlankPage	Remove blank pages from the output document. The default value is 'false'.												
int Sensitivity	The sensitivity (1 -100) for blank page detection. With high sensitivity, less blank pages are detected. Default value is 1.												
bool Deskew	Used to straighten the image. The default value is 'false'.												
int Despeckle	Removes specks below the specified pixel size from the image. The default value is 0 and the maximum value is 9.												

Setting	Description	
Interpolation	This class offers image interpolation features.	
	Property	Description
	bool Interpolate	Whether or not to interpolate. Default is 'false'.
	InterpolationMode InterpolationMode	Set the interpolation mode. Default is 'Normal'
	int TargetResolution	Interpolates the source image to the given resolution. This value must be greater than the source image's resolution.
bool KeepOriginalImage	Set this to true if you want to use the pre-processed image for OCR but keep the original image in the output document. Default value is 'true'.	

Setting	Description																												
LineRemoval	LineRemoval class provides line removal capabilities. The class contains algorithms for removing vertical and horizontal lines in an image. The integrator can control various settings in line removal algorithm like: • minimum and maximum line length and thickness of horizontal lines • minimum and maximum line length and thickness of vertical lines • clean noisy pixels adjacent to the line • maximum gap between the lines																												
	<table><tr><th>Property</th><th>Description</th></tr><tr><td>HorizontalCleanX</td><td>Cleans noisy pixels attached to horizontal lines. The default value is 1.</td></tr><tr><td>HorizontalCleanY</td><td>Cleans noisy pixels attached to horizontal lines. The default value is 1.</td></tr><tr><td>HorizontalDilate</td><td>Helps with the detection of horizontal lines. The default value is 1.</td></tr><tr><td>HorizontalMaxGap</td><td>The maximum horizontal line gap to close. It is useful to remove broken lines. The default value is 2.</td></tr><tr><td>HorizontalMaxThickness</td><td>The maximum thickness of horizontal lines to remove. It is useful to keep vertical lines larger than this parameter. Can also be useful to keep vertical letter strokes. The default value is 16.</td></tr><tr><td>HorizontalMinLength</td><td>The minimum length of horizontal lines to remove. The default value is 128.</td></tr><tr><td>RemoveLines</td><td>Whether or not to remove lines from an image. The default value is 'false'. <i>Note:</i> The image must be black and white for this setting to work. If you want to use this setting on a color image then you have to set 'Binarization' to 'true'.</td></tr><tr><td>VerticalCleanX</td><td>Cleans noisy pixels attached to vertical lines. The default value is 1.</td></tr><tr><td>VerticalCleanY</td><td>Cleans noisy pixels attached to vertical lines. The default value is 1.</td></tr><tr><td>VerticalDilate</td><td>Helps with the detection of vertical lines. The default value is 1.</td></tr><tr><td>VerticalMaxGap</td><td>The maximum vertical line gap to close. It is useful to remove broken lines. The default value is 2.</td></tr><tr><td>VerticalMaxThickness</td><td>The maximum thickness of vertical lines to remove. It is useful to keep horizontal lines larger than this parameter. Can also be useful to keep horizontal letter strokes. The default value is 16.</td></tr><tr><td>VerticalMinLength</td><td>The minimum length of vertical lines to remove. The default value is 128.</td></tr></table>	Property	Description	HorizontalCleanX	Cleans noisy pixels attached to horizontal lines. The default value is 1.	HorizontalCleanY	Cleans noisy pixels attached to horizontal lines. The default value is 1.	HorizontalDilate	Helps with the detection of horizontal lines. The default value is 1.	HorizontalMaxGap	The maximum horizontal line gap to close. It is useful to remove broken lines. The default value is 2.	HorizontalMaxThickness	The maximum thickness of horizontal lines to remove. It is useful to keep vertical lines larger than this parameter. Can also be useful to keep vertical letter strokes. The default value is 16.	HorizontalMinLength	The minimum length of horizontal lines to remove. The default value is 128.	RemoveLines	Whether or not to remove lines from an image. The default value is 'false'. <i>Note:</i> The image must be black and white for this setting to work. If you want to use this setting on a color image then you have to set 'Binarization' to 'true'.	VerticalCleanX	Cleans noisy pixels attached to vertical lines. The default value is 1.	VerticalCleanY	Cleans noisy pixels attached to vertical lines. The default value is 1.	VerticalDilate	Helps with the detection of vertical lines. The default value is 1.	VerticalMaxGap	The maximum vertical line gap to close. It is useful to remove broken lines. The default value is 2.	VerticalMaxThickness	The maximum thickness of vertical lines to remove. It is useful to keep horizontal lines larger than this parameter. Can also be useful to keep horizontal letter strokes. The default value is 16.	VerticalMinLength	The minimum length of vertical lines to remove. The default value is 128.
	Property	Description																											
	HorizontalCleanX	Cleans noisy pixels attached to horizontal lines. The default value is 1.																											
	HorizontalCleanY	Cleans noisy pixels attached to horizontal lines. The default value is 1.																											
	HorizontalDilate	Helps with the detection of horizontal lines. The default value is 1.																											
	HorizontalMaxGap	The maximum horizontal line gap to close. It is useful to remove broken lines. The default value is 2.																											
	HorizontalMaxThickness	The maximum thickness of horizontal lines to remove. It is useful to keep vertical lines larger than this parameter. Can also be useful to keep vertical letter strokes. The default value is 16.																											
	HorizontalMinLength	The minimum length of horizontal lines to remove. The default value is 128.																											
	RemoveLines	Whether or not to remove lines from an image. The default value is 'false'. <i>Note:</i> The image must be black and white for this setting to work. If you want to use this setting on a color image then you have to set 'Binarization' to 'true'.																											
	VerticalCleanX	Cleans noisy pixels attached to vertical lines. The default value is 1.																											
	VerticalCleanY	Cleans noisy pixels attached to vertical lines. The default value is 1.																											
	VerticalDilate	Helps with the detection of vertical lines. The default value is 1.																											
	VerticalMaxGap	The maximum vertical line gap to close. It is useful to remove broken lines. The default value is 2.																											
	VerticalMaxThickness	The maximum thickness of vertical lines to remove. It is useful to keep horizontal lines larger than this parameter. Can also be useful to keep horizontal letter strokes. The default value is 16.																											
VerticalMinLength	The minimum length of vertical lines to remove. The default value is 128.																												

Setting	Description
NonImagePDF NonImagePDF	This allows control over the treatment of non image-only PDFs, i.e. PDFs that have some text in them as well as images. The default value for this setting is 'OCR'.
PdfToImageBpp PdfToImageBpp	The Bits Per Pixel to use for the rasterized PDF page when using engine 1. This only applies for documents that are not processed using Native mode. The default value for this property is taken from the PDF page. <i>Note:</i> This should only be set with guidance from technical support.
PdfToImageCompression PdfToImageCompression	The compression to set to the images generated or rasterized from each page of the source PDF file. These images are then OCR'd to create the searchable PDF. The default value for this is taken from each page in the source PDF file.
PdfToImageDpi PdfToImageDpi	The DPI to set to the images rasterized from each page of the source PDF file. These images are then OCR'd to create the searchable PDF. The default value for this is taken from each page in the source PDF file. <i>Note:</i> This property is used in PdfToImageEngine 1 only.
int PdfToImageEngine	The engine to use for extracting or generating images from the source PDF file to use for OCRing. Valid values are 0 and 1.
bool 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. <i>Note:</i> This property is used in PdfToImageEngine 1 only.
int PdfToImageMaxRes	The maximum resolution of the rasterized images. If the resolution retrieved from the PDF page is bigger than this value, it will be set to this value. The default value for this property is 600. <i>Note:</i> This property is used in PdfToImageEngine 1 only.
int PdfToImageMinRes	The minimum resolution of the rasterized images. If the resolution retrieved from the PDF page is lower than this value, it will be set to this value. The default value for this property is 150. <i>Note:</i> This property is used in PdfToImageEngine 1 only.
bool RemoveDarkBorders	Removes the dark surrounding from bitonal, grayscale or color images. The dark surrounding of the image is whitened. <i>Note:</i> The dark border should be touching the edge of the image/page for this to work.

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

Setting	Description
AsianOCREngine AsianOCREngine	Set the Asian OCR engine to use when processing documents with Asian languages. Default is 'OcrAsian1'.
string Author	Set a custom Author in the output PDF document properties.
DateTime CreationDate	Set a custom creation date in the output PDF document properties. <i>Note:</i> If you are setting the date in through the properties.xml, then the date string should be in the format 'yyyy-MM-dd HH:mm:ss'.
bool EmbedFonts	Set the flag indicating whether to embed fonts in the output PDF document. <i>Note:</i> It is not recommended to embed fonts because it increases the file size. Unless you have some special needs, it is strongly advised to disable embedding fonts.
bool EnableConsoleOutput	If enabled then progress messages will be sent to the console. Default is true.
bool EnableCsvOutput	Enable or disable CSV file output.
bool EnableDebugOutput	If enabled then debug messages will be written to the console output.
bool EnableDocxOutput	Enable or disable DOCX file output.
bool EnableEpubOutput	Enable or disable EPUB file output. EPUB (short for electronic publication) is an e-book standard by the International Digital Publishing Forum (IDPF). Files have the extension .epub.
bool EnableExcelMIOutput	Enable or disable ExcelMIOutput file output.
bool EnableHtmlOutput	Enable or disable HTML file output.
bool EnableOpenDocumentTextOutput	Enable or disable Open document text file output.
bool EnablePdfOutput	Enable or disable PDF file output.
bool EnableRtfOutput	Enable or disable RTF file output.
bool EnableTextOutput	Enable or disable text file output.
bool EnableWordMIOutput	Enable or disable WordML file output.
bool EnableXlsxOutput	Enable or disable XLSX file output.
bool EnableXpsOutput	Enable or disable XPS file output.
int EndPage	Sets the last page of the source file that the OCR process will be run to (for a multipage source).
ExtractImageMethod ExtractImageMethod	This allows control over the method used to extract images from PDF files for OCR processing. The default value is 'Native'.
int GetPdfTextTimeout	The maximum time (in milliseconds) to check if a PDF document contains text or not.

Setting	Description
bool HandleExceptionsInternally	When set to true the OCR object will catch any exceptions for method calls and simply return false from the method. The exceptions caught are stored in the LastException property overwriting any previous value.

Setting	Description																												
IHQCCompression	This class manages the intelligent High Quality Compression parameters for PDF and XPS output formats. <i>Note:</i> You will need to have an IHQC license to use this option. <table> <tr> <th>Property</th><th>Description</th></tr> <tr> <td>int AdvancedBrightness</td><td></td></tr> <tr> <td>int AdvancedContrast</td><td></td></tr> <tr> <td>int AdvancedFactorPhoto</td><td></td></tr> <tr> <td>int AdvancedSmooth</td><td></td></tr> <tr> <td>bool AutoAdjust</td><td></td></tr> <tr> <td>bool EnableIHQCCompression</td><td>Apply Intelligent High Quality Compression. The default value is 'false'.</td></tr> <tr> <td>bool HighResolutionForeground</td><td>High resolution foreground feature enables a specific mode causing the use of high resolution for the foreground image. HighResolutionForeground enables a specific mode for working around a bug existing in the native readers of some platforms that does not read properly PDF (mainly the native reader on MAC OS X and IOS). Those native readers do not follow some of the specification of the PDF format. In place of displaying the correct compressed image, they display a down-sampled version of it. This special high resolution foreground mode offers a workaround to customer who wants to use iHQC with these buggy applications. Most of the PDF readers, including Adobe Reader are able to read PDF with IHQC compression properly on any platform.</td></tr> <tr> <td>IHQCCompressionLevel IHQCCompressionLevel</td><td>The compression level to be used. Level 1 is the basic compression level. Level 3 is the most advanced intelligent High Quality Compression mode. <i>Note:</i> PDF A-1a and PDF A-1b only supports level 1.</td></tr> <tr> <td>IHQCQualityFactor IHQCQualityFactor</td><td>Set the IHQC quality factor.</td></tr> <tr> <td>byte IHQCThreshold</td><td></td></tr> <tr> <td>int MinOutputResolution</td><td></td></tr> <tr> <td>SegmentationMode SegmentationMode</td><td></td></tr> <tr> <td>bool Smoothing</td><td></td></tr> </table>	Property	Description	int AdvancedBrightness		int AdvancedContrast		int AdvancedFactorPhoto		int AdvancedSmooth		bool AutoAdjust		bool EnableIHQCCompression	Apply Intelligent High Quality Compression. The default value is 'false'.	bool HighResolutionForeground	High resolution foreground feature enables a specific mode causing the use of high resolution for the foreground image. HighResolutionForeground enables a specific mode for working around a bug existing in the native readers of some platforms that does not read properly PDF (mainly the native reader on MAC OS X and IOS). Those native readers do not follow some of the specification of the PDF format. In place of displaying the correct compressed image, they display a down-sampled version of it. This special high resolution foreground mode offers a workaround to customer who wants to use iHQC with these buggy applications. Most of the PDF readers, including Adobe Reader are able to read PDF with IHQC compression properly on any platform.	IHQCCompressionLevel IHQCCompressionLevel	The compression level to be used. Level 1 is the basic compression level. Level 3 is the most advanced intelligent High Quality Compression mode. <i>Note:</i> PDF A-1a and PDF A-1b only supports level 1.	IHQCQualityFactor IHQCQualityFactor	Set the IHQC quality factor.	byte IHQCThreshold		int MinOutputResolution		SegmentationMode SegmentationMode		bool Smoothing	
Property	Description																												
int AdvancedBrightness																													
int AdvancedContrast																													
int AdvancedFactorPhoto																													
int AdvancedSmooth																													
bool AutoAdjust																													
bool EnableIHQCCompression	Apply Intelligent High Quality Compression. The default value is 'false'.																												
bool HighResolutionForeground	High resolution foreground feature enables a specific mode causing the use of high resolution for the foreground image. HighResolutionForeground enables a specific mode for working around a bug existing in the native readers of some platforms that does not read properly PDF (mainly the native reader on MAC OS X and IOS). Those native readers do not follow some of the specification of the PDF format. In place of displaying the correct compressed image, they display a down-sampled version of it. This special high resolution foreground mode offers a workaround to customer who wants to use iHQC with these buggy applications. Most of the PDF readers, including Adobe Reader are able to read PDF with IHQC compression properly on any platform.																												
IHQCCompressionLevel IHQCCompressionLevel	The compression level to be used. Level 1 is the basic compression level. Level 3 is the most advanced intelligent High Quality Compression mode. <i>Note:</i> PDF A-1a and PDF A-1b only supports level 1.																												
IHQCQualityFactor IHQCQualityFactor	Set the IHQC quality factor.																												
byte IHQCThreshold																													
int MinOutputResolution																													
SegmentationMode SegmentationMode																													
bool Smoothing																													

Setting	Description														
SupportedLanguages Language	Select the language to use for OCR processing. This will determine the dictionary that is used. If no language is set by the user, English will be used.														
SupportedLanguages[] 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. <i>Note:</i> Asian languages can't be mixed. Extended OCR SDK cannot load more than one Asian language at a time.														
Exception LastException	Stores last exception caught by the OCR object.														
Layout Layout	Select the layout to be used for docx and rtf output. The default value is 'Standard'.														
string License	Specifies the license key														
LicenseInfo	Get the current license information. <i>Note:</i> This needs to be called after setting <code>Ocr.License</code>. <table border="1"> <thead> <tr> <th>Property</th><th>Description</th></tr> </thead> <tbody> <tr> <td>bool AsianOCREnabled</td><td>Get whether Asian OCR is enabled.</td></tr> <tr> <td>bool IHQCEnabled</td><td>Get whether Intelligent High Quality Compression (IHQC) is enabled.</td></tr> <tr> <td>bool IsTrial</td><td>Get whether the license is trial</td></tr> <tr> <td>string LicenseType</td><td>Get the license type: Basic, Standard, Advanced or Extended.</td></tr> <tr> <td>int LicenseType</td><td>Get the maximum number of threads allowed to run concurrently</td></tr> <tr> <td>string SDKVersion</td><td>Get the version of the SDK that this license is for.</td></tr> </tbody> </table>	Property	Description	bool AsianOCREnabled	Get whether Asian OCR is enabled.	bool IHQCEnabled	Get whether Intelligent High Quality Compression (IHQC) is enabled.	bool IsTrial	Get whether the license is trial	string LicenseType	Get the license type : Basic, Standard, Advanced or Extended.	int LicenseType	Get the maximum number of threads allowed to run concurrently	string SDKVersion	Get the version of the SDK that this license is for.
Property	Description														
bool AsianOCREnabled	Get whether Asian OCR is enabled.														
bool IHQCEnabled	Get whether Intelligent High Quality Compression (IHQC) is enabled.														
bool IsTrial	Get whether the license is trial														
string LicenseType	Get the license type : Basic, Standard, Advanced or Extended.														
int LicenseType	Get the maximum number of threads allowed to run concurrently														
string SDKVersion	Get the version of the SDK that this license is for.														
string LogFilePath	Gets the path of the log file where output is written to if 'LogToFile' property is set to true. <i>Note:</i> The value for this property must be accessed after calling 'Ocr.Recognize'.														
bool LogToFile	Set this to true if you want to log output to a file. The file will be stored in the TempFolder specified.														
DateTime ModifiedDate	Set a custom modified date in the output PDF document properties. <i>Note:</i> If you are setting the date in through the properties.xml, then the date string should be in the format 'yyyy-MM-dd HH:mm:ss'.														

Setting	Description										
bool NoOcr	Set this to true if you do not want to OCR the source document. This could be useful if you want to convert a TIFF to an image-only PDF.										
int NumberPages	Returns the number pages in a document.										
PDFImageCompression	The compression parameters to use for color JPEG images in generated PDFs. Note: This won't be used if you set ExtractImageMethod to 'Native' because the 'Native' method does not make any changes to the PDF file. <table> <tr> <th>Property</th><th>Description</th></tr> <tr> <td>bool EnableJpeg2000Compression</td><td>Use JPEG 2000 compression. If set to true, then the default compression mode is JPEG2000_QUALITY_FACTOR. The default value is false.</td></tr> <tr> <td>Jpeg2000CompressionMode Jpeg2000CompressionMode</td><td>The JPEG 2000 compression mode to use.</td></tr> <tr> <td>int Jpeg2000CompressionValue</td><td>The value to use for the selected compression mode.</td></tr> <tr> <td>byte QualityFactor</td><td>Gets/Sets the image compression quality factor of color JPEG images in generated PDFs. The default value is 192. The quality factor determines the degree of loss caused by the compression process. Extended OCR accepts a value from 0 to 255: 255 guarantees the highest image quality 0 guarantees the best compression</td></tr> </table>	Property	Description	bool EnableJpeg2000Compression	Use JPEG 2000 compression. If set to true, then the default compression mode is JPEG2000_QUALITY_FACTOR. The default value is false.	Jpeg2000CompressionMode Jpeg2000CompressionMode	The JPEG 2000 compression mode to use.	int Jpeg2000CompressionValue	The value to use for the selected compression mode.	byte QualityFactor	Gets/Sets the image compression quality factor of color JPEG images in generated PDFs. The default value is 192. The quality factor determines the degree of loss caused by the compression process. Extended OCR accepts a value from 0 to 255: 255 guarantees the highest image quality 0 guarantees the best compression
Property	Description										
bool EnableJpeg2000Compression	Use JPEG 2000 compression. If set to true, then the default compression mode is JPEG2000_QUALITY_FACTOR. The default value is false.										
Jpeg2000CompressionMode Jpeg2000CompressionMode	The JPEG 2000 compression mode to use.										
int Jpeg2000CompressionValue	The value to use for the selected compression mode.										
byte QualityFactor	Gets/Sets the image compression quality factor of color JPEG images in generated PDFs. The default value is 192. The quality factor determines the degree of loss caused by the compression process. Extended OCR accepts a value from 0 to 255: 255 guarantees the highest image quality 0 guarantees the best compression										
int PdfToImageExtractionTimeout	The maximum time (in milliseconds) to take to get image(s) from a PDF page. Default value is 100,000 ms.										
PDFVersion PDFVersion	The PDF version of the resulting output document. The default value is 1.4. Note: This won't be used if you set ExtractImageMethod to 'Native' because the 'Native' method does not make any changes to the PDF file, except if you choose to convert to PDFAb.										
int ProcessPageTimeout	The maximum time (in milliseconds) to take to process a page (apply pre-processing and OCR). Default value is 300,000 ms.										
bool ReadBmpViaFreeImage	Contact technical support (support@aquaforest.com) for guidance on using this property.										

Setting	Description
bool RemoveExistingPDFText	This applies only when a PDF is being used as the source for OCR. When set to true this will not include any searchable text that already exists from the source document. Such functionality might be useful if the source document was created by OCR of an image only PDF or other image file and the quality of the text from the previous OCR is poor. Note: There is no way to distinguish text added as a result of OCR from text added by other means and as a result this option should be used with care.
bool RetainBookmarks	Retains any bookmarks from the source file in the output when using ExtractImageMethod.ConvertToTiff
bool RetainCreationDate	Retains the creation date of the source file in the output PDF document properties.
bool RetainMetadata	Retains any metadata from the source file in the output when using ExtractImageMethod.ConvertToTiff
bool RetainModifiedDate	Retains the modified date of the source file in the output PDF document properties.
int StartPage	Sets the first page of the source file that the OCR process will be begin from (for a multipage source).
string TempFolder	The temporary folder used to keep the intermediary processing files.
int ThreadCount	The number of worker threads that are used for performing page recognition. Please note that, in order to get the best performance, the number of worker threads need to be set accordingly to your system capabilities. Aquaforest Extended OCR provides an "automatic" mode for balancing the number of work threads based on the system where it is executed. In order to activate this mode, set this value to 0.
string Version	Returns the version of the SDK being used.
byte WorkDepth	This property can have values going from 0 to 255: 0 meaning "best speed" and 255 "best accuracy". Increasing the work depth allows the OCR engine to spend more time on low-quality documents. The "work depth" indicates how deep the engine is allowed to work in order to find a satisfactory result. For documents of good quality, the engine will find a satisfactory result at an early stage and there will not be much speed differences between "best speed" and "best accuracy". For bad quality documents, if allowed by "word depth", the engine will work deeper. The accuracy will be better but at the price of speed.

5.2.3 Methods

Method	Description
void Abort()	Terminates processing.
void ConvertPDFToPDFa(string source, string target, PDFVersion version)	Converts a PDF file to PDF/A without going through the OCR process.
void DeleteTemporaryFiles()	Delete temporary files used in the OCR processing

Method	Description
Image GetPageImage(int pageNumber)	Retrieves the image for the specified page. <i>Note:</i> Make sure that you call this method before deleting temporary files.
void ReadBMPSource(string sourcefile)	Read a bitmap source for OCR.
void ReadBMPSource(Stream source)	Read a bitmap source from a System.IO.Stream source for OCR.
string ReadDocumentString()	Returns a string containing the words from all pages processed.
void ReadImageSource(Image image)	Reads an Image object checking the number of frames (pages).
void ReadJPEGSource(string sourcefile)	The source JPEG file to OCR.
void ReadJPEGSource(Stream source)	Read a JPEG source from a System.IO.Stream source for OCR.
string ReadPageString(int pageNumber)	This function will retrieve a string representing the words found for the page requested. <i>Note:</i> This should be called only after OCR has completed successfully for the page.
string ReadPageString(int pageNumber, Rectangle region)	This function will retrieve a string representing the words found for the page requested and in the region specified. <i>Note:</i> This should be called only after OCR has completed successfully for the page.
Words ReadPageWords(int pageNumber)	This function will retrieve a Words object representing the words found for the page requested. <i>Note:</i> This should be called only after OCR has completed successfully for the page.
Words ReadPageWords(int pageNumber, Rectangle region)	This function will retrieve a Words object representing the words found for the page requested and in the region specified. <i>Note:</i> This should be called only after OCR has completed successfully for the page.
void ReadPDFSource(string sourcefile)	The source PDF file to OCR.
void ReadPDFSource(Stream source)	Read a PDF source from a System.IO.Stream source for OCR.
void ReadPDFSource(string sourcefile, string password)	The password protected source PDF file to OCR.
void ReadPDFSource(Stream source, string password)	Read a password protected PDF source from a System.IO.Stream source for OCR.
void ReadTIFFSource(string sourcefile)	The source TIFF file to OCR.
void ReadTIFFSource(Stream source)	Read a TIFF source from a System.IO.Stream source for OCR.
bool Recognize(PreProcessor preProcessor)	Call Recognize to perform the actual OCR once options have been set on the Ocr and PreProcessor objects.

Method	Description
bool SaveCSVOutput(string outputfile, bool overwriteExisting)	Save CSV file to the specified output location. <i>Note:</i> This function should only be called if EnableCsvOutput was set to true prior to the call to Recognize.
bool SaveCSVOutput(out Stream outputStream)	Save CSV output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableCsvOutput was set to true prior to the call to Recognize.
bool SaveDOCXOutput(string outputfile, bool overwriteExisting)	Save DOCX file to the specified output location. <i>Note:</i> This function should only be called if EnableDocxOutput was set to true prior to the call to Recognize.
bool SaveDOCXOutput (out Stream outputStream)	Save DOCX output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableDocxOutput was set to true prior to the call to Recognize.
bool SaveEPUBOutput(string outputfile, bool overwriteExisting)	Save EPUB file to the specified output location. <i>Note:</i> This function should only be called if EnableEpubOutput was set to true prior to the call to Recognize.
bool SaveEPUBOutput (out Stream outputStream)	Save EPUB output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableEpubOutput was set to true prior to the call to Recognize.
bool SaveExcelMLOutput(string outputfile, bool overwriteExisting)	Save Excel Markup Language file to the specified output location. <i>Note:</i> This function should only be called if EnableExcelMLOutput was set to true prior to the call to Recognize.
bool SaveExcelMLOutput (out Stream outputStream)	Save Excel Markup Language output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableExcelMLOutput was set to true prior to the call to Recognize.
bool SaveHTMLOutput(string outputfile, bool overwriteExisting)	Save HTML file to the specified location. <i>Note:</i> This function should only be called if EnableHtmlOutput was set to true prior to the call to Recognize.
bool SaveHTMLOutput (out Stream outputStream)	Save HTML output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableHtmlOutput was set to true prior to the call to Recognize.
bool SaveOpenDocumentTextOutput(string outputfile, bool overwriteExisting)	Save Open Document Text file to the specified output location. <i>Note:</i> This function should only be called if EnableOpenDocumentTextOutput was set to true prior to the call to Recognize.

Method	Description
bool SaveOpenDocumentTextOutput (out Stream outputStream)	Save Open Document Text output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableOpenDocumentTextOutput was set to true prior to the call to Recognize.
bool SavePDFOutput(string outputfile, bool overwriteExisting)	Save PDF file to the specified output location. <i>Note:</i> This function should only be called if EnablePdfOutput was set to true prior to the call to Recognize.
bool SavePDFOutput (out Stream outputStream)	Save PDF output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnablePdfOutput was set to true prior to the call to Recognize.
bool SaveRTFOutput(string outputfile, bool overwriteExisting)	Save RTF file to the specified output location. <i>Note:</i> This function should only be called if EnableRtfOutput was set to true prior to the call to Recognize.
bool SaveRTFOutput (out Stream outputStream)	Save RTF output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableRtfOutput was set to true prior to the call to Recognize.
bool SaveTextOutput(string outputfile, bool overwriteExisting)	Save Text file to the specified output location. <i>Note:</i> This function should only be called if EnableTextOutput was set to true prior to the call to Recognize.
bool SaveTextOutput (out Stream outputStream)	Save Text output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableTextOutput was set to true prior to the call to Recognize.
bool SaveWordMLOutput(string outputfile, bool overwriteExisting)	Save Word Markup Language file to the specified output location. <i>Note:</i> This function should only be called if EnableWordMLOutput was set to true prior to the call to Recognize.
bool SaveWordMLOutput (out Stream outputStream)	Save Word Markup Language output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableWordMLOutput was set to true prior to the call to Recognize.
bool SaveXLSXOutput(string outputfile, bool overwriteExisting)	Save XLSX file to the specified output location. <i>Note:</i> This function should only be called if EnableXlsxOutput was set to true prior to the call to Recognize.
bool SaveXLSXOutput (out Stream outputStream)	Save XLSX output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableXlsxOutput was set to true prior to the call to Recognize.

Method	Description
bool SaveXPSOutput(string outputfile, bool overwriteExisting)	Save XPS file to the specified output location. <i>Note:</i> This function should only be called if EnableXpsOutput was set to true prior to the call to Recognize.
bool SaveXPSOutput (out Stream outputStream)	Save XPS output to a System.IO.Stream. <i>Note:</i> This function should only be called if EnableXpsOutput was set to true prior to the call to Recognize.

5.2.4 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.

5.2.5 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
int Count	Returns the number of WordData objects in the collection.
int Height	Returns the height of the current word.
int Width	Returns the width of the current word.

5.3.6 Words 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.

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

Enumeration	Description																														
AsianOCREngine	The Asian OCR Engine to use when processing documents with Asian languages. <table><tr><th>Name</th><th>Value</th></tr><tr><td>OcrAsian1</td><td>0</td></tr><tr><td>OcrAsian2</td><td>1</td></tr></table>	Name	Value	OcrAsian1	0	OcrAsian2	1																								
Name	Value																														
OcrAsian1	0																														
OcrAsian2	1																														
ExtractImageMethod	Whether to convert the images in a PDF document to TIFF or not. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Native</td><td>0</td><td>This method places the OCR'd text directly into a copy of the original PDF rather than creating an entirely new PDF.</td></tr><tr><td>ConvertToTiff</td><td>1</td><td>The PDF is rasterized to create an image of each page in the PDF.</td></tr></table>	Name	Value	Description	Native	0	This method places the OCR'd text directly into a copy of the original PDF rather than creating an entirely new PDF.	ConvertToTiff	1	The PDF is rasterized to create an image of each page in the PDF.																					
Name	Value	Description																													
Native	0	This method places the OCR'd text directly into a copy of the original PDF rather than creating an entirely new PDF.																													
ConvertToTiff	1	The PDF is rasterized to create an image of each page in the PDF.																													
IHQCCompressionLevel	The IHQC compression level. Note: PDF A-1a and PDF A-1b only supports level 1. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>LevelI</td><td>1</td><td>Level I (Acrobat 5.0 and higher)</td></tr><tr><td>Level2a</td><td>2</td><td>Level II spec a</td></tr><tr><td>Level2b</td><td>3</td><td>Level II spec b</td></tr><tr><td>Level3</td><td>4</td><td>Level III (Acrobat 6.0 and higher)</td></tr></table>	Name	Value	Description	LevelI	1	Level I (Acrobat 5.0 and higher)	Level2a	2	Level II spec a	Level2b	3	Level II spec b	Level3	4	Level III (Acrobat 6.0 and higher)															
Name	Value	Description																													
LevelI	1	Level I (Acrobat 5.0 and higher)																													
Level2a	2	Level II spec a																													
Level2b	3	Level II spec b																													
Level3	4	Level III (Acrobat 6.0 and higher)																													
IHQCQualityFactor	The IHQC quality factor. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>MinimalSize</td><td>1</td><td>Minimal size</td></tr><tr><td>VerySmallSize</td><td>2</td><td>Very small size</td></tr><tr><td>SmallSize</td><td>3</td><td>Small size</td></tr><tr><td>FavorSizeOverQuality</td><td>4</td><td>Favor size over quality</td></tr><tr><td>Medium</td><td>5</td><td>Medium</td></tr><tr><td>FavorQualityOverSize</td><td>6</td><td>Favor quality over size</td></tr><tr><td>HighQuality</td><td>7</td><td>High quality</td></tr><tr><td>VeryHighQuality</td><td>8</td><td>Very high quality</td></tr><tr><td>MaximalQuality</td><td>9</td><td>Maximal quality</td></tr></table>	Name	Value	Description	MinimalSize	1	Minimal size	VerySmallSize	2	Very small size	SmallSize	3	Small size	FavorSizeOverQuality	4	Favor size over quality	Medium	5	Medium	FavorQualityOverSize	6	Favor quality over size	HighQuality	7	High quality	VeryHighQuality	8	Very high quality	MaximalQuality	9	Maximal quality
Name	Value	Description																													
MinimalSize	1	Minimal size																													
VerySmallSize	2	Very small size																													
SmallSize	3	Small size																													
FavorSizeOverQuality	4	Favor size over quality																													
Medium	5	Medium																													
FavorQualityOverSize	6	Favor quality over size																													
HighQuality	7	High quality																													
VeryHighQuality	8	Very high quality																													
MaximalQuality	9	Maximal quality																													

Enumeration	Description															
InterpolationMode	Interpolation algorithms. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Normal</td><td>0</td><td>Normal interpolation mode.</td></tr><tr><td>Fast</td><td>1</td><td>Fast interpolation mode.</td></tr></table>	Name	Value	Description	Normal	0	Normal interpolation mode.	Fast	1	Fast interpolation mode.						
Name	Value	Description														
Normal	0	Normal interpolation mode.														
Fast	1	Fast interpolation mode.														
Jpeg2000CompressionMode	The JPEG 2000 compression mode. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>JPEG2000_QUALITY_FACTOR</td><td>0</td><td>Use quality factor.</td></tr><tr><td>JPEG2000_TARGET_FILE_SIZE</td><td>1</td><td>Use target file size.</td></tr><tr><td>JPEG2000_COMPRESSION_RATIO</td><td>2</td><td>Use compression ratio.</td></tr></table>	Name	Value	Description	JPEG2000_QUALITY_FACTOR	0	Use quality factor.	JPEG2000_TARGET_FILE_SIZE	1	Use target file size.	JPEG2000_COMPRESSION_RATIO	2	Use compression ratio.			
Name	Value	Description														
JPEG2000_QUALITY_FACTOR	0	Use quality factor.														
JPEG2000_TARGET_FILE_SIZE	1	Use target file size.														
JPEG2000_COMPRESSION_RATIO	2	Use compression ratio.														
Layout	The layout to use when generating non-pdf documents such as DOCX and RTF. <table><tr><th>Name</th><th>Value</th></tr><tr><td>Standard</td><td>0</td></tr><tr><td>Flow</td><td>1</td></tr></table>	Name	Value	Standard	0	Flow	1									
Name	Value															
Standard	0															
Flow	1															
NonImagePDF	This allows control over the treatment of non image-only PDFs, i.e. PDFs that have some text in them as well as images. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>OCR</td><td>0</td><td>The document will be OCRed using the image extraction method defined by "ExtractImageMethod".</td></tr><tr><td>RaiseError</td><td>1</td><td>The task will terminate with an error.</td></tr><tr><td>Skip</td><td>2</td><td>The document will not be processed.</td></tr><tr><td>PassThrough</td><td>3</td><td>The file will not be processed, but a copy of the document will be made and named as if the processing had occurred.</td></tr></table>	Name	Value	Description	OCR	0	The document will be OCRed using the image extraction method defined by "ExtractImageMethod".	RaiseError	1	The task will terminate with an error.	Skip	2	The document will not be processed.	PassThrough	3	The file will not be processed, but a copy of the document will be made and named as if the processing had occurred.
Name	Value	Description														
OCR	0	The document will be OCRed using the image extraction method defined by "ExtractImageMethod".														
RaiseError	1	The task will terminate with an error.														
Skip	2	The document will not be processed.														
PassThrough	3	The file will not be processed, but a copy of the document will be made and named as if the processing had occurred.														
PDFAVersion	The PDF/A version when using ConvertPDFToPDF/A method. <table><tr><th>Member name</th><th>Value</th><th>Description</th></tr><tr><td>PDF_A1b</td><td>2</td><td>PDF/A-1b</td></tr><tr><td>PDF_A2b</td><td>6</td><td>PDF/A-2b</td></tr><tr><td>PDF_A3b</td><td>11</td><td>PDF/A-3b</td></tr></table>	Member name	Value	Description	PDF_A1b	2	PDF/A-1b	PDF_A2b	6	PDF/A-2b	PDF_A3b	11	PDF/A-3b			
Member name	Value	Description														
PDF_A1b	2	PDF/A-1b														
PDF_A2b	6	PDF/A-2b														
PDF_A3b	11	PDF/A-3b														

Enumeration	Description																																							
PdfToImageBpp	The Bits Per Pixel to use for the rasterized PDF page. This only applies for documents that are not processed using Native mode. <table><tr><th>Name</th><th>Value</th></tr><tr><td>Bpp_1</td><td>1</td></tr><tr><td>Bpp_24</td><td>24</td></tr></table>	Name	Value	Bpp_1	1	Bpp_24	24																																	
Name	Value																																							
Bpp_1	1																																							
Bpp_24	24																																							
PDFToImageCompression	The compression to use for the rasterized PDF page. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>CCITT4</td><td>CCITT4</td><td>CCITT4 compression schema</td></tr><tr><td>LZW</td><td>LZW</td><td>LZW compression schema</td></tr></table>	Name	Value	Description	CCITT4	CCITT4	CCITT4 compression schema	LZW	LZW	LZW compression schema																														
Name	Value	Description																																						
CCITT4	CCITT4	CCITT4 compression schema																																						
LZW	LZW	LZW compression schema																																						
PDFVersion	Set the PDF Version of the output file. Note: This will only affect documents that are not processed natively and documents that are converted to PDF/A. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>PDF_1_4</td><td>0</td><td>PDF 1.4 - 2001 - Acrobat 5.0 + JBIG2; RC4 encryption key lengths greater than 40bits (128bits); Extensible Metadata Platform (XMP); Tagged PDF</td></tr><tr><td>PDF_A1b</td><td>1</td><td>PDF 1.4 A-1b</td></tr><tr><td>PDF_A1a</td><td>2</td><td>PDF 1.4 A-1a</td></tr><tr><td>PDF_1_5</td><td>3</td><td>PDF 1.5 - 2003 - Acrobat 6.0 + JPEG 2000</td></tr><tr><td>PDF_1_6</td><td>4</td><td>PDF 1.6 - 2005 - Acrobat 7.0 + AES encryption</td></tr><tr><td>PDF_1_7</td><td>5</td><td>PDF 1.7 - 2006 - Acrobat 8.0 -> 2008 - ISO 32000-1:2008</td></tr><tr><td>PDF_A2b</td><td>6</td><td>PDF 1.7 A-2b</td></tr><tr><td>PDF_A2a</td><td>7</td><td>PDF 1.7 A-2a</td></tr><tr><td>PDF_1_7_3</td><td>8</td><td>PDF 1.7 - 2008 - Adobe Extension Level 3 / Acrobat 9.0 + 256-bit AES encryption</td></tr><tr><td>PDF_1_7_5</td><td>9</td><td>PDF 1.7 - 2009 - Adobe Extension Level 5 / Acrobat 9.1</td></tr><tr><td>PDF_1_7_8</td><td>10</td><td>PDF 1.7 - 2011 - Adobe Extension Level 8 / Acrobat X</td></tr><tr><td>PDF_A3b</td><td>11</td><td>PDF A-3b</td></tr></table>	Name	Value	Description	PDF_1_4	0	PDF 1.4 - 2001 - Acrobat 5.0 + JBIG2; RC4 encryption key lengths greater than 40bits (128bits); Extensible Metadata Platform (XMP); Tagged PDF	PDF_A1b	1	PDF 1.4 A-1b	PDF_A1a	2	PDF 1.4 A-1a	PDF_1_5	3	PDF 1.5 - 2003 - Acrobat 6.0 + JPEG 2000	PDF_1_6	4	PDF 1.6 - 2005 - Acrobat 7.0 + AES encryption	PDF_1_7	5	PDF 1.7 - 2006 - Acrobat 8.0 -> 2008 - ISO 32000-1:2008	PDF_A2b	6	PDF 1.7 A-2b	PDF_A2a	7	PDF 1.7 A-2a	PDF_1_7_3	8	PDF 1.7 - 2008 - Adobe Extension Level 3 / Acrobat 9.0 + 256-bit AES encryption	PDF_1_7_5	9	PDF 1.7 - 2009 - Adobe Extension Level 5 / Acrobat 9.1	PDF_1_7_8	10	PDF 1.7 - 2011 - Adobe Extension Level 8 / Acrobat X	PDF_A3b	11	PDF A-3b
Name	Value	Description																																						
PDF_1_4	0	PDF 1.4 - 2001 - Acrobat 5.0 + JBIG2; RC4 encryption key lengths greater than 40bits (128bits); Extensible Metadata Platform (XMP); Tagged PDF																																						
PDF_A1b	1	PDF 1.4 A-1b																																						
PDF_A1a	2	PDF 1.4 A-1a																																						
PDF_1_5	3	PDF 1.5 - 2003 - Acrobat 6.0 + JPEG 2000																																						
PDF_1_6	4	PDF 1.6 - 2005 - Acrobat 7.0 + AES encryption																																						
PDF_1_7	5	PDF 1.7 - 2006 - Acrobat 8.0 -> 2008 - ISO 32000-1:2008																																						
PDF_A2b	6	PDF 1.7 A-2b																																						
PDF_A2a	7	PDF 1.7 A-2a																																						
PDF_1_7_3	8	PDF 1.7 - 2008 - Adobe Extension Level 3 / Acrobat 9.0 + 256-bit AES encryption																																						
PDF_1_7_5	9	PDF 1.7 - 2009 - Adobe Extension Level 5 / Acrobat 9.1																																						
PDF_1_7_8	10	PDF 1.7 - 2011 - Adobe Extension Level 8 / Acrobat X																																						
PDF_A3b	11	PDF A-3b																																						

Enumeration	Description																																																									
SegmentationMode	IHQC segmentation modes. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>Color</td><td>0</td><td>IHQC segmentation only with input color image.</td></tr><tr><td>BWhite_and_color</td><td>1</td><td>IHQC segmentation with dual stream images.</td></tr><tr><td>Dualstream</td><td>2</td><td>IHQC segmentation with input color image and input black and white image.</td></tr></table>	Name	Value	Description	Color	0	IHQC segmentation only with input color image.	BWhite_and_color	1	IHQC segmentation with dual stream images.	Dualstream	2	IHQC segmentation with input color image and input black and white image.																																													
Name	Value	Description																																																								
Color	0	IHQC segmentation only with input color image.																																																								
BWhite_and_color	1	IHQC segmentation with dual stream images.																																																								
Dualstream	2	IHQC segmentation with input color image and input black and white image.																																																								
SupportedLanguages	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. <table><tr><th>Name</th><th>Code</th><th>Description</th></tr><tr><td>English</td><td>0</td><td>English (American)</td></tr><tr><td>German</td><td>1</td><td></td></tr><tr><td>French</td><td>2</td><td></td></tr><tr><td>Spanish</td><td>3</td><td></td></tr><tr><td>Italian</td><td>4</td><td></td></tr><tr><td>British</td><td>5</td><td></td></tr><tr><td>Swedish</td><td>6</td><td></td></tr><tr><td>Danish</td><td>7</td><td></td></tr><tr><td>Norwegian</td><td>8</td><td></td></tr><tr><td>Dutch</td><td>9</td><td></td></tr><tr><td>Portuguese</td><td>10</td><td></td></tr><tr><td>Brazilian</td><td>11</td><td></td></tr><tr><td>Galician</td><td>12</td><td></td></tr><tr><td>Icelandic</td><td>13</td><td></td></tr><tr><td>Greek</td><td>14</td><td></td></tr><tr><td>Czech</td><td>15</td><td></td></tr><tr><td>Hungarian</td><td>16</td><td></td></tr><tr><td>Polish</td><td>17</td><td></td></tr></table>	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	
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																																																									

Enumeration	Description		
	Romanian	18	
	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

Enumeration	Description	
		to use this language.
	Tchinese	50 You will need to have the Asian OCR license 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

Enumeration	Description		
	Tahitian	82	
	Tonga	83	
	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	
	Hebrew	105	
	Numeric	114	This language limits recognition to numeric characters.
	Esperanto	115	
	Maltese	116	
	Zulu	117	
	Afaan	118	
	Asturian	119	
	AzeriLatin	120	
	Luba	121	
	Papamiento	122	

Enumeration	Description		
	Tatar	123	
	Turkmen	124	
	Welsh	125	
	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: ocr.PDFVersion = PDFVersion.PDF_1_4_Ab
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 Methods

Aquaforest	Extended
ConfigurePDFStamp	n/a

6.1.5 PreProcessor Properties

Aquaforest	Extended
n/a	AdvancedDespeckle
Binarize	Binarization

Aquaforest	Extended
BlackPixelLimit	n/a
n/a	BlankPageRemoval
BlankPageThreshold	BlankPageRemoval.Sensitivity
BoxSize	n/a
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:

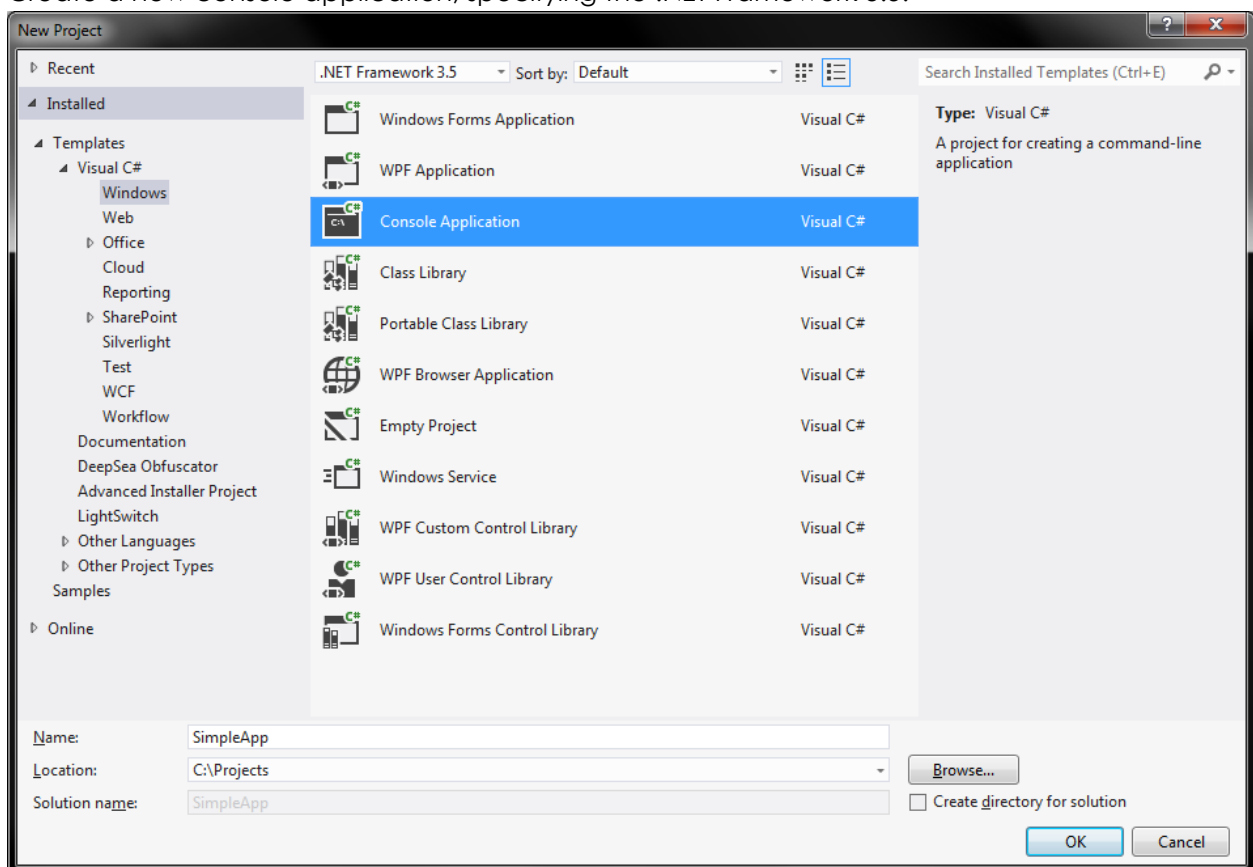
- display the rotation of each page
- identify whether each page contains text or not
- recognize text in different languages

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

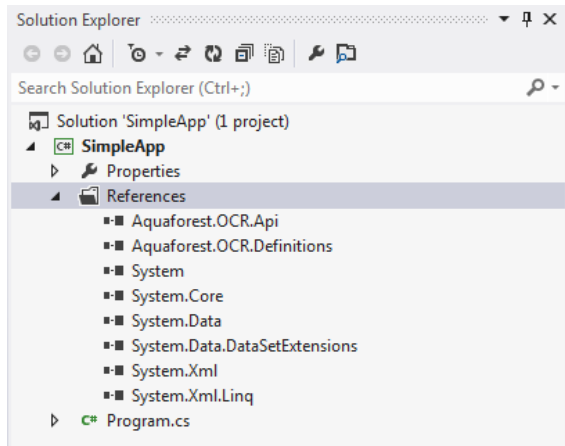
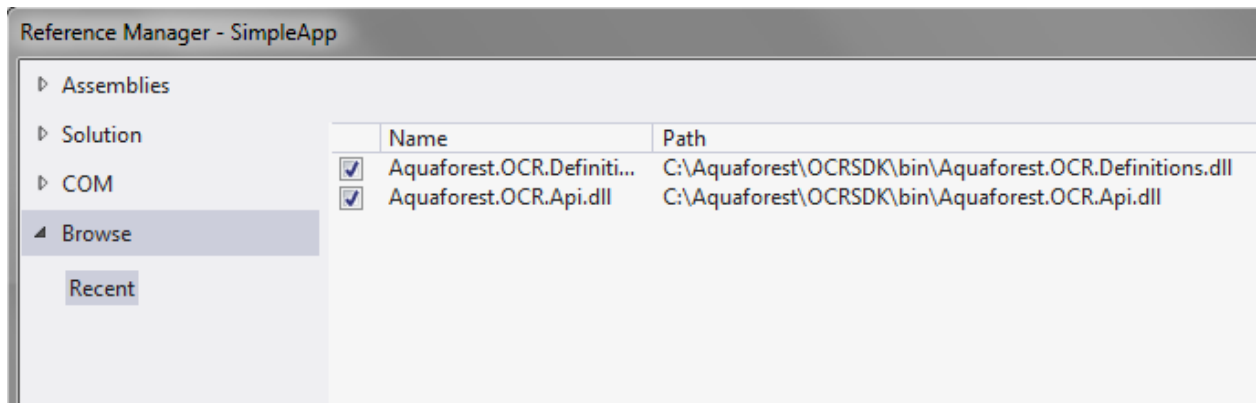
- Page 1 and 4 contains text in English
- Page 2 contains no text
- Page 3 contains text in 4 different languages: English, French, Spanish and German
- Page 5 is rotated 180 °
- Page 6 contains text in French

6.2.1 Using Aquaforest SDK and Visual Studio 2012

1. Create a new console application, specifying the .NET Framework 3.5.



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;

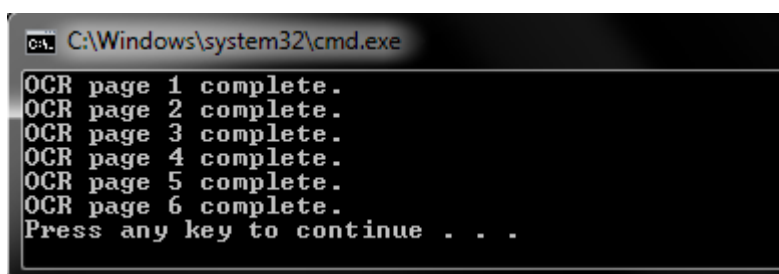
                ocr.ReadTIFFSource(@"C:\MyFiles\input\sample.tif");

                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:

```
C:\Windows\system32\cmd.exe

Page 1
Contains Text: True
Rotation: 0
OCR page 1 complete.
Could not recognise text on page 2
Attempting to reprocess the page as image only.
-----
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.

EN	FR	ES	DE								
Using the Remote Use the remote to control compatible media phone players. <table><tr><td>Answer/End Call</td><td>Click the center button once.</td></tr></table> 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	Answer/End Call	Click the center button once.	Utilisation de la télécommande Utiliser la télécommande pour commander les lecteurs multimédia sur téléphones compatibles. <table><tr><td>Réponse / Fin d'appel</td><td>Cliquer une fois sur le bouton central</td></tr></table> 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 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	Réponse / Fin d'appel	Cliquer une fois sur le bouton central	Uso del control remoto Usate il telecomando per agire sul vostro lettore multimediale compatibile. <table><tr><td>Invio/ Fine chiamata</td><td>Pulse el botón central una vez</td></tr></table> Quando rispondete alle chiamate, parlate normalmente. 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 – è possibile che su alcuni telefoni sia necessario regolare il volume quando si passa dalla telefonata all'ascolto di musica. Para más información sobre modelos compatibles, acuda a: www.shure.com	Invio/ Fine chiamata	Pulse el botón central una vez	Verwendung der Fernsteuerung Die Fernsteuerung zur Bedienung kompatibler Medienwiedergabegeräte/ Handys verwenden. <table><tr><td>Anruf annehmen/ beenden</td><td>Mittlere Taste einmal anklicken.</td></tr></table> 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 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	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:

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	Utilisation de la télécommande Utiliser la télécommande pour commander les lecteurs multimedia sur telephones compatibles. Rspone / 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.	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 — è possibile che su alcuni telefoni sia necessario regolare il volume quando si passa dalla telefonata all'ascolto di musica.	Verwendung der Fernsteuerung Die Fernsteuerung zur Bedienung kompatibler Medienwiedergabegeräte/ 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 gedrückt 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
---	--	---	---

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"
- "reconnaître" 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]:

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	Utilisation de la telecommande Utiliser la telecommande pour commander les lecteurs multimedia sur telephones compatibles. Rponse / 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.	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—è possibile che su alcuni telefoni sia necessario regolare il volume quando si passa dalla telefonata all'ascolto di musica.	Verwendung der Fernsteuerung Die Fernsteuerung zur Bedienung kompatibler Medienwiedergabegeräte/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 gedrückt 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
--	---	---	---

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

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 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. 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.
--

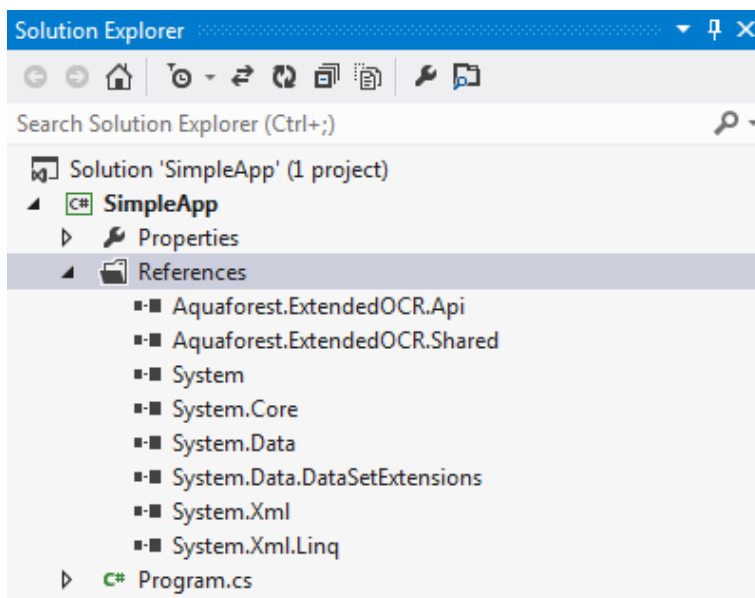
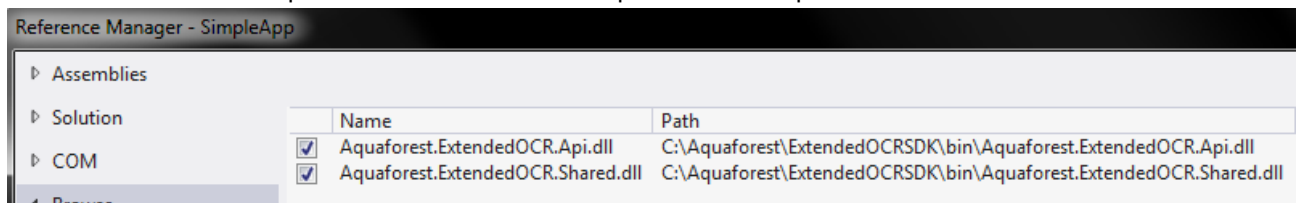
If we analyse the OCR results of page 6, we can see that all the French accents are now recognised properly. However, if we examine the OCR results of page 3, we'll notice that the SDK still did not recognise the special characters of the other languages. It actually produced the same results as the 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

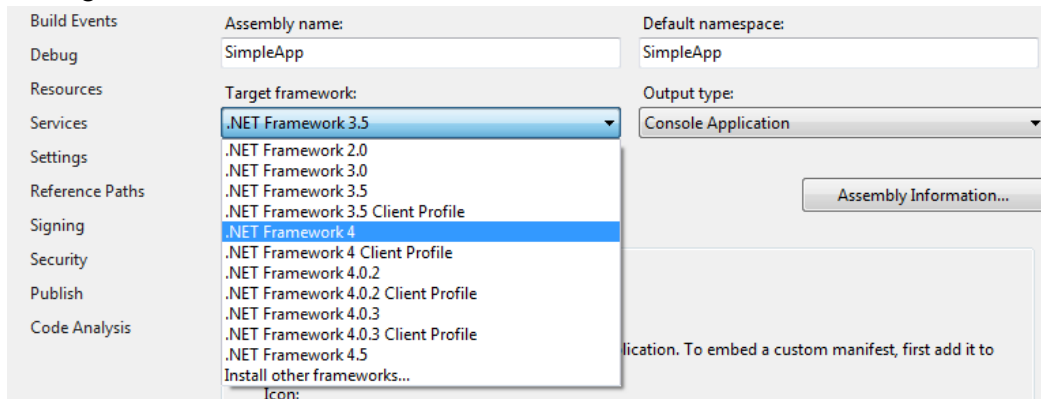
This section explains how to change the application created in the previous section so that it uses the Extended SDK module.

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. Change the .NET Framework from 3.5 to 4:



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:

```

C:\Windows\system32\cmd.exe
Extended OCR 2.0.140408.0 - Aquaforest Extended OCR Trial License
Loading...C:\MyFiles\input\sample.tif (6) pages
Processing Page 1
-----
Page 1
Contains Text: True
Rotation: 0
Processing Page 2
-----
Page 2
Contains Text: False
Rotation: 0
Processing Page 3
-----
Page 3
Contains Text: True
Rotation: 0
Processing Page 4
-----
Page 4
Contains Text: True
Rotation: 0
Processing Page 5
-----
Page 5
Contains Text: True
Rotation: 180
Processing Page 6
-----
Page 6
Contains Text: True
Rotation: 0
Creating PDF file...
Total elapsed time: 0m 10s 886ms
Press any key to continue . . .

```

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)

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 button, check for additional features or to troubleshoot usage problems.	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 de l'appareil mobile pour trouver de plus amples renseignements	Uso del control remoto Usate il telecomando per agire sui vostro lettore multimediale compatibile. Invio/ I Pulse el botón Fine central una vez chiamata Quando rispondete alle chiamate, parlate normalmente. I comandi Invio/Fine chiamata su lia maggior parte di telefoni vengono effettuati mediante pressione del pulsante centrale. Per ulteriori informazioni sull'uso del pulsante di invio/fine chiamata, su lia ricerca di funzioni aggiuntive o sull'individuazione di problemi d'uso, leggete	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 weitere Informationen über die Verwendung der Taste Annehmen/Beenden sowie über sonstige
--	--	--	---

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	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.	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	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
---	--	---	--

Page 6 OCR results (with Extended module)

Éditeurs 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

7.2.2 .NET Framework

.NET Version 3.5

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 3.5	Visual Studio 2008

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();
            }
        }
    }
}
```

8 Barcode Module API Reference

8.1 BarcodeReader Class

This class is used to decode barcodes inside images or PDF documents.

8.1.1 Constructor

- `BarcodeReader reader = new BarcodeReader(settings);`
"settings" is an instance of the [BarcodeReaderSettings](#) class.
- `BarcodeReader reader = new BarcodeReader(settingsList);`
"settingsList" is a List of [BarcodeReaderSettings](#) (`List<BarcodeReaderSettings>`)

8.1.2 Properties

Property	Description
int BlankPageThreshold	Use this to set the minimum number of "On Pixels" that must be present in the image for a page not to be considered blank. A value of -1 will turn off blank page detection. The default value is -1.
Dictionary<int, DecodeResult > DecodeResults	Returns the decoded results in a dictionary, where the dictionary key is the page number and the dictionary value is the DecodeResult .
bool EnableConsoleOutput	If enabled then progress messages will be sent to the console. Default is true.
bool EnableDebugOutput	If enabled then debug messages will be written to the console output. Default is false.
int EndPage	Sets the last page of the source file where barcode recognition will end (for a multipage source).
string LastErrorMessage	Stores the last error message caught by the BarcodeReader object.
Exception LastException	Stores the last Exception caught by the BarcodeReader object.
string License	Specifies the license key.
string LogFilePath	The path of the log file where output is written to if LogToFile property is set to true.
bool LogToFile	Set this to true if you want to log output to a file. The file will be stored in the LogFilePath if specified or the TempFolder if not.
bool OverwriteExisting	Set this to true if you want to overwrite any existing files when splitting by barcode.
PdfToImageCompression PdfToImageCompression	The compression to set to the TIFF images generated or converted from the source PDF file. These images are then OCR'd to create the searchable PDF.
PdfToImageDpi PdfToImageDpi	The DPI of TIFF images generated from the source PDF file. These images are then used for barcode recognition. The default value for this is taken from each page in the source PDF file.
bool 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.
int PdfToImageMaxRes	The minimum resolution to use when saving the extracted images from a PDF file. The default value for this is 600.

int PdfToImageMinRes	The minimum resolution to use when saving the extracted images from a PDF file. The default value for this is 200.
bool PerformPreprocessing	When set to true, the barcode reader will perform pre-processing on the image before barcode recognition is performed. The pre-processing settings are found in the BarcodeReaderSettings supplied in the constructor of BarcodeReader.
bool Pinvoke	Set this to true if you want to launch the pre-processing process through pinvoke.
string ResourceFolder	The folder containing all the assemblies required for barcode recognition. If the SDK is installed in the default location, this value is 'C:\Aquaforest\OCRSdk\bin'.
bool SplitByBarcode	Set this to true to split multi-page PDF or TIFF documents by barcode identified on the pages.
string SplitInvalidPathCharSubstitute	If the barcode value that is used to replace %VALUE% in SplitOutputPath contains invalid path characters, they will be replaced by the value set in this property. The default is a space ' '.
SplitMode SplitMode	The split mode to use.
string SplitNoBarcodeTemplate	The renaming template to use for page ranges where no barcodes were identified. Allowed templates: - %INDEX% Replaced by the current split index. - %FILENAME Replaced by the filename of the source file. Not applicable for System.IO.Stream and System.Drawing.Image sources. The default value is %FILENAME%_NO_BARCODE_%INDEX%
string SplitOutputPath	The output file path template where the split files will be saved. Example path: C:\Output\%VALUE%_%INDEX%.pdf: - %VALUE% Replaced by the barcode value found. - %INDEX% Replaced by the current split index. - %FILENAME Replaced by the filename of the source file. Not applicable for System.IO.Stream and System.Drawing.Image sources. If the path does not contain any of the above templates, the %INDEX% template will be added to the path.
int StartPage	Sets the first page of the source file where barcode recognition will begin from (for a multipage source).
string TempFolder	The temporary folder used to keep the intermediary processing files.

8.1.3 Methods

Method	Description
bool DecodeImage(Image image)	Scans a System.Drawing.Image for barcodes. Returns true if at least one barcode is found.

Method	Description
bool DecodeImage(Stream source)	Scans an image System.IO.Stream for barcodes. Returns true if at least one barcode is found.
bool DecodeImage(String sourceImage)	Scans the specified source image file for barcodes. Returns true if at least one barcode is found.
bool DecodePdf(Stream source)	Scans a PDF System.IO.Stream for barcodes. Returns true if at least one barcode is found.
bool DecodePdf(String sourcePdf)	Scans the specified PDF file for barcodes. Returns true if at least one barcode is found.
bool DecodePdf(Stream source, String password)	Scans a password protected PDF System.IO.Stream for barcodes. Returns true if at least one barcode is found.
bool DecodePdf(string sourcePdf, string password)	Scans the specified password protected PDF file for barcodes. Returns true if at least one barcode is found.
Dictionary<int, DecodeResult > GetAllDecodeResults()	Get all the decoded results in a dictionary, where the dictionary key is the page number and the dictionary value is the DecodeResult .
DecodeResult GetDecodeResult(int pageNumber)	Get all the decoded result(s) of a particular page.

8.2 BarcodeReaderSettings Class

This class contains all the barcode and pre-processing settings used for decoding barcodes.

8.2.1 Constructor

[BarcodeReaderSettings](#) settings = new [BarcodeReaderSettings](#)();

8.2.2 Properties

Property	Description
BarcodeFormats BarcodeFormats	The barcode formats to search for. Set formats as follows: - BarcodeFormats[BarcodeFormat.All_1D] = true or false; - BarcodeFormats[BarcodeFormat.AZTEC] = true or false; - etc...
int Binarize	This value should generally only be used under guidance from technical support. It can control the way that color images are processed and force binarization with a particular threshold. A value of 200 has been shown to generally give good results in testing, but this should be confirmed with "typical" customer documents. By setting this to -1 an alternative method is used which will attempt to separate the text from any background images or colors. This can give improved results for certain documents such as newspaper and magazine pages. <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.

float BlackPixelLimit	Contact technical support (support@aquaforest.com) for guidance on using this property. <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
int BoxSize	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 barcode reader. Technically, this option removes connected elements with a minimum area (in pixels and defined by this property). This option is currently only applied for bitonal images. <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
bool Deskew	Deskew (straighten) the image. The default value is false (disabled). <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
int Despeckle	Despeckle the image – The method removes all disconnected elements within the image that have height or width in pixels less than the specified figure. The maximum value is 9 and the default value is 0. <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
int GrayscaleQuality	This option should generally only be used under guidance from technical support (support@aquaforest.com). <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
bool LibTiffSavePageAsBmp	Sometimes if there is an image which is 1bpp and has LZW compression, the pre-processing can cause the colour of the image to be inverted (black to white and white to black). Set this to true to avoid this. <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
float MaxDeskewAngle	Maximum angle by which a page will be deskewed. <i>Note:</i> This option should generally only be used under guidance from technical support (support@aquaforest.com). <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
float MinDeskewConfidence	This option should generally only be used under guidance from technical support (support@aquaforest.com). <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.

string Morph	Morphological options that will be applied to the binarized image before OCR. If set to empty 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. This option should generally only be used under guidance from technical support. <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
bool ProductsOnly	Only decode the UPC and EAN families of barcodes.
bool ReadMultipleBarcodes	Scans each image/page for multiple barcodes.
string RemoveLines	Remove lines from images. Typical value is 100.5. This option should generally only be used under guidance from technical support (support@aquaforest.com). <i>Note:</i> BarcodeReader.PerformPreprocessing needs to be set to true for this to work.
bool TryHarder	Spend more time to try to find a barcode; optimize for accuracy, not speed. Default is true.
Rectangle Zone	Only examine the region specified for barcode(s).

8.3 BarcodeResult Class

This class encapsulates the result of decoding a barcode within an image.

8.3.1 Properties

Property	Description
BarcodeFormat BarcodeFormat	Represents the format of the barcode that was decoded.
byte[] RawBytes	Raw bytes encoded by the barcode, if applicable, otherwise null.
List< BarcodeResultPoint > ResultPoints	Points related to the barcode in the image. These are typically points identifying finder patterns or the corners of the barcode. The exact meaning is specific to the type of barcode that was decoded.
string Text	Raw text encoded by the barcode, if applicable, otherwise null.
long Timestamp	Gets the timestamp.
BarcodeResultType Type	The type of data encoded by a barcode.

8.4 BarcodeResultPoint Class

This class encapsulates a point of interest in an image containing a barcode. Typically, this would be the location of a finder pattern or the corner of the barcode, for example.

8.4.1 Properties

Property	Description
float X	The X co-ordinate.
float Y	The Y co-ordinate.

8.5 DecodeResult Class

This class encapsulates the decoded result of each page/image.

8.5.1 Properties

Property	Description
BarcodeResult BarcodeResult	The barcode result. If BarcodeReaderSettings. ReadMultipleBarcodes is set to True, the results will be in BarcodeResults .
List< BarcodeResult > BarcodeResults	The barcode results. If ReadMultipleBarcodes is set to False, the result will be in BarcodeResult .
string ErrorMessage	Any errors that occurred during the decoding of this image/page.
bool IsBlankPage	Whether the page is blank. This value is only set when BlankPageThreshold is greater than -1.
int PageNumber	Gets the current page number.
bool Success	Whether a barcode was found on the page.

8.6 Enumerations

Enumeration	Description		
BarcodeFormat	Barcode formats supported.		
	Name	Value	Description
	AZTEC	1	Aztec 2D barcode format.
	CODABAR	2	CODABAR 1D format.
	CODE_39	4	Code 39 1D format.
	CODE_93	8	Code 93 1D format.
	CODE_128	16	Code 128 1D format.
	DATA_MATRIX	32	Data Matrix 2D barcode format.
	EAN_8	64	EAN-8 1D format.
	EAN_13	128	EAN-13 1D format.
	ITF	256	ITF (Interleaved Two of Five) 1D format (Code 25).
	MAXICODE	512	MaxiCode 2D barcode format.
	PDF_417	1024	PDF417 format.
	QR_CODE	2048	QR Code 2D barcode format.
	RSS_14	4096	RSS 14
	RSS_EXPANDED	8192	RSS EXPANDED
	UPC_A	16384	UPC-A 1D format.
	UPC_E	32768	UPC-E 1D format.
	UPC_EAN_EXTENSION	65536	UPC/EAN extension format. Not a stand-alone format.
	MSI	131072	MSI
	PLESSEY	262144	Plessey
	All_1D	61918	UPC_A UPC_E EAN_13 EAN_8 CODABAR CODE_39 CODE_93 CODE_128 ITF RSS_14 RSS_EXPANDED without MSI (too many false-positives)

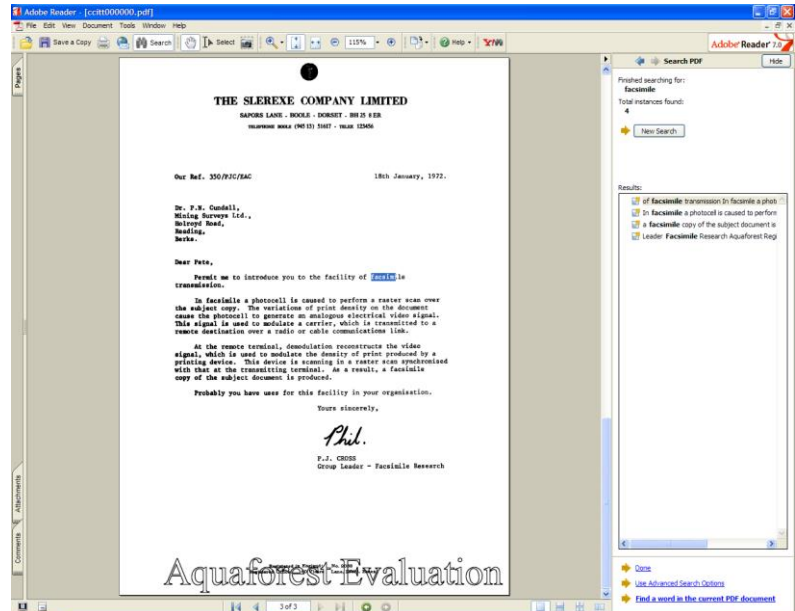
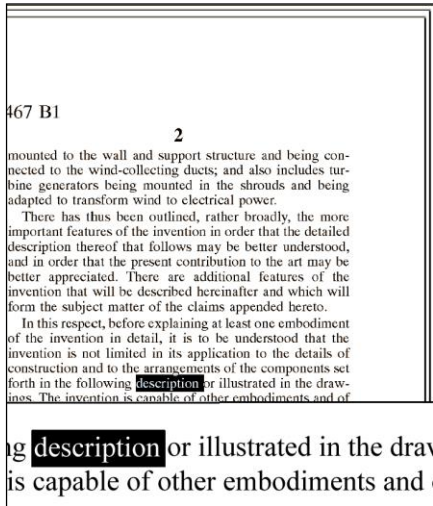
Enumeration	Description																										
BarcodeResultType	Represents the type of data encoded by a barcode. <table><tr><th>Name</th><th>Value</th></tr><tr><td>ADDRESSBOOK</td><td>0</td></tr><tr><td>EMAIL_ADDRESS</td><td>1</td></tr><tr><td>PRODUCT</td><td>2</td></tr><tr><td>URI</td><td>3</td></tr><tr><td>TEXT</td><td>4</td></tr><tr><td>GEO</td><td>5</td></tr><tr><td>TEL</td><td>6</td></tr><tr><td>SMS</td><td>7</td></tr><tr><td>CALENDAR</td><td>8</td></tr><tr><td>WIFI</td><td>9</td></tr><tr><td>ISBN</td><td>10</td></tr><tr><td>VIN</td><td>11</td></tr></table>	Name	Value	ADDRESSBOOK	0	EMAIL_ADDRESS	1	PRODUCT	2	URI	3	TEXT	4	GEO	5	TEL	6	SMS	7	CALENDAR	8	WIFI	9	ISBN	10	VIN	11
Name	Value																										
ADDRESSBOOK	0																										
EMAIL_ADDRESS	1																										
PRODUCT	2																										
URI	3																										
TEXT	4																										
GEO	5																										
TEL	6																										
SMS	7																										
CALENDAR	8																										
WIFI	9																										
ISBN	10																										
VIN	11																										
PdfToImageDpi	The DPI to use for images extracted or rasterized from each PDF page. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>DPI_150</td><td>150</td><td>150 dpi</td></tr><tr><td>DPI_200</td><td>200</td><td>200 dpi</td></tr><tr><td>DPI_300</td><td>300</td><td>300 dpi</td></tr><tr><td>DPI_400</td><td>400</td><td>400 dpi</td></tr><tr><td>DPI_500</td><td>500</td><td>500 dpi</td></tr><tr><td>DPI_600</td><td>600</td><td>600 dpi</td></tr></table>	Name	Value	Description	DPI_150	150	150 dpi	DPI_200	200	200 dpi	DPI_300	300	300 dpi	DPI_400	400	400 dpi	DPI_500	500	500 dpi	DPI_600	600	600 dpi					
Name	Value	Description																									
DPI_150	150	150 dpi																									
DPI_200	200	200 dpi																									
DPI_300	300	300 dpi																									
DPI_400	400	400 dpi																									
DPI_500	500	500 dpi																									
DPI_600	600	600 dpi																									
PDFToImageCompression	The compression to use for the rasterized PDF page. <table><tr><th>Name</th><th>Value</th><th>Description</th></tr><tr><td>CCITT4</td><td>CCITT4</td><td>CCITT4 compression schema</td></tr><tr><td>LZW</td><td>LZW</td><td>LZW compression schema</td></tr></table>	Name	Value	Description	CCITT4	CCITT4	CCITT4 compression schema	LZW	LZW	LZW compression schema																	
Name	Value	Description																									
CCITT4	CCITT4	CCITT4 compression schema																									
LZW	LZW	LZW compression schema																									

Enumeration	Description
SplitMode	Split modes.

9 Background - Searchable PDFs

9.1 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.



9.2 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.

9.3 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.

9.3.1 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.

9.3.2 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.

9.3.3 Despeckle

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

9.3.4 Deskew

This option can improve OCR results by straightening crooked pages.

9.3.5 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.

9.3.6 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).

9.3.7 Language Settings

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

9.4 Hardware and Performance

9.4.1 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.

9.4.2 Exploiting Multiple CPUs

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

9.4.3 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.

10 Acknowledgements

This product makes use of a number of Open Source components which are included in binary form. The appropriate acknowledgements and copyright notices are given below.

LEPTONICA

Copyright (C) 2001 Leptonica. All rights reserved.

LIBJPEG

This software is based in part on the work of the Independent JPEG Group.

ZLIB

(C) 1995-2004 Jean-loup Gailly and Mark Adler.

ITEX 4.1.6

Copyright (C) 1999-2009 by Bruno Lowagie and Paulo Soares et al. All Rights Reserved. Binaries distributed under the Mozilla Public License.

CUNEIFORM

Copyright (c) 1993-2008, Cognitive Technologies. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Cognitive Technologies nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

LIBTIFF

Copyright (c) 1988-1997 Sam Leffler. Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

FREEIMAGE

This software uses the FreeImage open source image library. See <http://freeimage.sourceforge.net> for details. FreeImage is used under the (FIPL), version 1.0.

ZXING

This software uses the ZXing open source barcode library. See <https://github.com/zxing/zxing/wiki/License-Questions> for details. ZXing is used under the Apache Version 2.0 license: <http://www.apache.org/licenses/LICENSE-2.0.html>