Aquaforest

Aquaforest SDK 3.0
Reference Guide

Aquaforest

Making Content Findable

Aquaforest SDK
Reference Guide

Version 3.0

October 2020

© Aquaforest Ltd 2001 - 2020

Content

1 INTRODUCTION 5

1.1 SDK Overview 5
1.1.1 Licensing 5
1.1.2 Folders 5

1.2 System Requirements 6
1.2.1 Supported Environments 6
1.2.2 .NET Framework 6
1.2.3 Visual C++ Runtime 6

1.3 Application Deployment 6
1.3.1 Prerequisites 6
1.3.2 Deploying your application 6
1.3.3 Upgrading existing Aquaforest SDK solutions 7

1.4 Technical Support 7

2 DATA EXTRACTOR MODULE 7

2.1 Overview 7

2.2 Folders 7

2.3 Application Development and Deployment 7
2.3.1 References 7
2.3.2 Licensing 7
2.3.3 Deploying C# and VB.NET Applications 8
2.3.4 Recognition Engine Class 8
2.3.4.1 Methods and Properties 8
2.3.4.2 Custom Keys 8
2.3.4.3 Expected Keys and Synonyms 8
2.3.5 Properties.xml 9

3 STANDARD OCR MODULE 11

3.1 Overview 11

3.2 Folders 11

3.3 Application Development and Deployment 11
3.3.1 References 11
3.3.2 Licensing 11

3.4 Standard OCR API 11
3.4.1 PreProcessor Class 11
3.4.2 Ocr Class 12
3.4.3 StatusUpdateEventArgs Class 12
3.4.4 Words Class 13
3.4.5 WordData Class 13
3.4.6 PdfMerger Class 14
3.4.7 Disposal and Temporary Files folders 14

3.4.8 Multi-threaded applications 14
3.4.9 Words Class 14
3.4.10 Properties File 14

4 EXTENDED OCR MODULE 16

4.1 Overview 16

4.2 Folders 17

4.3 Application Development and Deployment 17
4.3.1 References 17
4.3.2 Properties.xml 17

4.4 Extended OCR API 18
4.4.1 Classes 18
4.4.2 Methods and Properties 18
4.4.3 Events 19
4.4.3.1 StatusUpdateEventArgs Class 19
4.4.3.2 Words Class 19
4.4.3.3 WordData Class 20
4.4.3.4 CharacterData Class 20

4.5 Supported Languages 21
4.5.1 Language Table 22

4.6 Fonts 24

5 CLOUD OCR MODULE 25

5.1 Overview 25
5.1.1 Microsoft Computer Vision 25
5.1.2 Google Cloud Vision 26

5.2 Folders 26

5.3 Application Development and Deployment 26
5.3.1 References 26
5.3.2 Licensing 26
5.3.3 Classes 26
5.3.4 Engine Settings 27
5.3.4.1 MicrosoftCloudOCR Properties 27
5.3.4.2 GoogleCloudOcr Properties 28
5.3.4.3 CloudOcr Properties 29

5.4 Image Requirements 29
5.4.1 Microsoft Image Requirements 29
5.4.2 Google Image Requirements 30

5.5 Fonts 30

6 PDF TOOLKIT MODULE 31

6.1 Overview 31

6.2 Folders 31

6.3 PDF Toolkit API 31

6.3.1 API Samples 31
6.3.2 Product License Key 32
6.3.3 API Documentation 32
6.3.4 Deployment 32
6.3.4.1 C# and VB Deployment 32

7 BARCODE RECOGNITION MODULE 33

7.1 Overview 33
7.1.1 Supported Barcode Formats 33

7.2 Folders 33

7.3 Application Development and Deployment 33
7.3.1 References 33

7.4 Barcode Module API 33

8 PROCESS LOGGING 34

8.1 Logging Types 34
8.1.1 SimpleConsoleLogger 34
8.1.2 SimpleFileLogger 34
8.1.3 Custom Logger 34

8.2 Logging Levels 34

8.3 Usage 34

9 ACKNOWLEDGEMENTS 36

1 Introduction

1.1 SDK Overview

Aquaforest SDK for .NET applications combines capabilities from all our existing SDK programs, and also introduces our newly developed Data Extractor module. This module is a highly requested product that allows data extraction from PDF documents without the need for templates or prior training. The software is able to read the PDF text and extract important key-value pairs automatically, making processing of files with various layouts easy.

Other modules enable developers to directly make use of the OCR Engines, giving full control over OCR processes in their custom applications. It also contains components for manipulating PDFs and recognizing barcodes.

The SDK consists of 6 modules:

Standard OCR Engine – Generate fully text searchable PDFs with the original image and a transparent text layer. Uses customizable pre-processor setting for optimal OCR output.

Extended OCR Engine – Utilises the IRIS engine for benefits over the standard engine, including enhanced recognition, increased language support and PDF version support.

Cloud OCR Engine – Supports the use of the OCR engines from both Microsoft and Google, which excels at handwriting recognition. Before using these, you will need a subscription.

PDF Toolkit – Contains a set of command line tools for creating, processing and manipulating PDF files.

Barcode Recognition – Supports decoding barcodes from both image and PDF files, supporting a wide variety of barcode formats.

Data Extraction – Automatically retrieves key-value pairs from documents. Users can declare specific keys to manipulate the extracted data more easily.

In addition there is an Aquaforest Logging class which provides an optional process logging facility, this can log to the System Console, a file or a combination of the two. See section 8 for details

1.1.1 Licensing

Aquaforest SDK is licensed differently to our other products, allowing buyers to choose which modules they will use. Send an email to support@aquaforest.com to discuss your options.

1.1.2 Folders

The "**Aquaforest SDK**" main folder contains the following folder structure:

- **Barcode**
  - **bin** - Contains all the assemblies, DLLs and configurations files for Barcode projects
  - **samples** – Contains Barcode samples in C#
- **Data Extractor**
  - **bin** - Contains all the assemblies, DLLs and configurations files for Data Extractor projects
  - **samples** – Contains Data Extractor samples in C# and VB.NET
- **Diagnostics**
  - Contains a pre-requisite checker to identify any missing requirements
- **docs**
  - Contains documentation for using the SDK
- **License**
  - Contains the Aquaforest License Agreement
- **OCR**
  - **Cloud**
    - **bin** - Contains all the assemblies, DLLs and configurations files for Cloud OCR projects
    - **samples** – Contains Cloud OCR samples in C#
  - **Extended**

        - **bin** - Contains all the assemblies, DLLs and configurations files for Extended OCR projects
        - **samples** – Contains Extended OCR samples in C#, VB.NET and ASP.NET
    - **Standard**
        - **bin** - Contains all the assemblies, DLLs and configurations files for Standard OCR projects
        - **samples** – Contains Standard OCR samples in C#, VB.NET and ASP.NET
- **PDF Toolkit**
    - **bin** - Contains all the assemblies, DLLs and configurations files for PDF Toolkit projects
    - **samples** – Contains PDF Toolkit samples in C# and VB.NET
- **welcome**
    - Contains the SDK welcome page

The main folder also contains a shortcut to the SDK welcome page, named "**Aquaforest SDK 3.0**". The "**bin**" folders are explained further in the "**Deployment Guide**", identifying the files that need to be referenced and listing the files that need to be in the folder for deployment.

1.2 System Requirements

1.2.1 Supported Environments

- Windows 8
- Windows 10
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2016
- Windows Server 2019

1.2.2 .NET Framework

.NET Version 4.7.2

1.2.3 Visual C++ Runtime

The Visual C++ 2017 Redistributable package is required for development as well as deployment.

1.3 Application Deployment

1.3.1 Prerequisites

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

Application Platform	VC++ Redistributable	Minimum .NET Framework Version	Minimum Visual Studio Version
x86	Visual C++ Redistributable 2017 x86	.NET Framework 4.7.2	Visual Studio 2017
x64	Visual C++ Redistributable 2017 x86 Visual C++ Redistributable 2017 x64		
Any CPU	Visual C++ Redistributable 2017 x86 Visual C++ Redistributable 2017 x64		

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

1.3.2 Deploying your application

Please see the Deployment Guide for detailed instructions.

Any deployment method should ensure that the target system meets the requirements and has the Visual C++ 2017 Redistributable package and .NET Version 4.7.2 framework.

Each module has its own bin folder, and the contents of those files will be required as part of the deployment.

1.3.3 Upgrading existing Aquaforest SDK solutions

If you are upgrading from a previous Aquaforest development kit (PDF Toolkit or OCR SDK), a number of minor changes will be required to your solutions to make use of the new combined SDK.

Previous versions of the development kits will need to be uninstalled before Aquaforest SDK 3 can be installed.

There are details on upgrading solutions (based on upgrading solutions from previous versions of the development kits) in the Upgrade Guide.

1.4 Technical Support

Twelve months Support and Maintenance is included in the purchase price. Support and Maintenance cover can optionally be renewed after 12 months.

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 Data Extractor Module

2.1 Overview

The Aquaforest Data Extractor module gives applications the capability of extracting important data from PDF files without knowing or needing to be trained on the layout of the information.

Data is automatically extracted from the document as key-value pairs, the module handles the identification of keys and their associated values, without the need to declare specific extraction zones.

Custom keys allow localization in both language and format with associated value types having specific confidence levels.

Expected keys allow named values to be extracted and the associated synonyms allow variations in wording/spelling to be handled in the background.

2.2 Folders

The Aquaforest Data Extractor Module contains the following folders:

- bin – Contains the binaries used by the Data Extractor module
- samples – contains samples (in C# and VB) illustrating how to make use of the Data Extractor module

2.3 Application Development and Deployment

2.3.1 References

To use the Data Extractor API, a reference to `Aquaforest.DataExtractor.Api` must be added in your application. This will allow you to use methods to retrieve key/value pairs from documents in your program.

You must also have the files **DPDFRenderNative_x86.dll** (32-bit) and/or **DPDFRenderNative_x64.dll** (64-bit) in your output file path location. These files are necessary to the functioning of the Recognition Engine and can be copied from the Recognition Engine bin folder.

2.3.2 Licensing

Production system deployment requires that a license string is defined in the code. The license string decides the number of concurrent processes that can be run. For the Data Extractor module, you pass the license key to the constructor.

For example:

```
string license = "<your-license-key>";
RecognitionEngine recognitionEngine = new RecognitionEngine(license, resourcePath, logger);
```

2.3.3 Deploying C# and VB.NET Applications

Ensure that the target system meets the system requirements described in section 1.2.

In most environments, your project's Build Output folder will contain the required files.

2.3.4 Recognition Engine Class

The `RecognitionEngine` object is used to perform recognition on image and PDF files, producing key/value pairs as data output. You can access modifiable settings through the `Settings` field, and set the expected key to be returned from the extraction process.

This class also contains the results returned from extraction in the form of both the expected key/values and the full results. There are multiple methods to return or save output using different output formats, such as JSON and CSV.

2.3.4.1 Methods and Properties

Refer to the 'Aquaforest.DataExtractor' section of the "**Aquaforest SDK 3.0.chm**" file found in the folder "[SDK installation path]\docs\help" to view all the properties and methods available for the Recognition Engine class.

2.3.4.2 Custom Keys

If you know that a certain string will be a key, you can pre-define it as a custom key, making it more easily identified as a key during the recognition process. You can also associate one or more data types for the value and give each data type a confidence value. During the extraction process the potential values found are identified as particular data types and the confidence value is used as part of the process of selecting the best value.

For example, the custom key "Balance Due" has two potential value data types:

- "Number" with a confidence value of 80
- "Currency" with a confidence value of 100

If a potential "Balance Due" key was found in a document, then the associated value would more likely be one that was identified as Currency than a Number.

This also allows language localization by adding the translated custom key. For documents in Welsh, a custom key could be added called "Balans sy'n ddyledus" with the same value data types as "Balance Due".

Custom keys can be added from files. A default custom key file can be defined in the "**Properties.xml**" file under the `<CustomKeysFilePaths>` field. There is a pre-existing custom keys file found in this location "[SDK installation path]\DataExtractor\bin\resources\CustomKeys.json".

The custom key can be plain text or a regular expression, and you can also define the expected datatype of the associated value, giving each datatype a confidence score. This file must be in the correct JSON format. Use the **CustomKeys.json** as a template when adding new custom keys.

2.3.4.3 Expected Keys and Synonyms

Expected keys are used to define named key-value pairs that contain data required for the application. This makes their use in applications extremely easy.

As an example, an expected key in invoice documents might be "Invoice No".

Once a document has been recognized, this value (if found) can be referenced by:

```
recognitionEngine.GetExpectedKeyValue("Invoice Number");
```

There are many potential variations in spelling or wording of "Invoice Number" keys, such as: "Inv No"; "Invoice No" etc. To make handling these variations easy, expected keys can also have synonyms. These synonyms could cover variations in spelling or wording, or even languages.

Expected keys and their optional synonyms are both handled through the `Settings` field of `RecognitionEngine` class.

Default keys and synonyms can be loaded from files listed in the `<ExpectedKeysFilePaths>` field of the **Properties.xml** file

Additional expected keys and synonyms can also be added through code, either directly or from additional files.

For example:

```
recognitionEngine.Settings.ExpectedKeys.Add("Invoice No", "Invoice Number");
recognitionEngine.Settings.ExpectedKeys.AddFromFile(expectedKeysPath);
```

For more details see the "Aquaforest.DataExtractor" section of the "**Aquaforest SDK 3.0.chm**".

The format of the JSON files is as follows:

```
{
  "expectedKeys": [
      {
        "expectedKey": "Invoice No",
        "synonyms": [
        "Invoice Number",
        "Invoice Num"
        ]
      },
      {
        "expectedKey": "Inv Date",
        "synonyms": [
        "Invoice Date",
        "Inv. Date",
        "Inv date"
        ]
      }
  ]
}
```

At the top level, "expectedKeys" is a set of "expectedKey" and "synonyms" pairs.

Each "expectedKey" is a string that the recognition engine will try to find an associated value for in the processed document.

The expected key may not be exactly the same each time, so a set of synonyms can be associated with them. If no value is found for the expected key, a value will be taken from a synonym if available.

There is no limit to the "expectedKey" or "synonyms" entries. If there are no synonyms, the square brackets can be left empty.

2.3.5 Properties.xml

The **Properties.xml** file contains the base settings that are loaded whenever the `RecognitionEngine` class is instantiated, transferring its values to the `Settings` field. In addition to these settings, the file also contains other fields that control the logging output and default file paths for the expected and custom keys.

Field	Notes
EnableConsoleOutput	Displays the Data Extractor's logging output on the console
EnableDebugOutput	Sets the logging level to Debug, giving more in depth logging output
LogToFile	Creates logs in the output file path when enabled
LoadDefaultCustomKeys	If set to true, custom keys will be loaded from paths in the "CustomKeysFilePaths"
CustomKeysFilePaths	The paths to json files containing the default Custom Keys
IgnoreCaseCustomKeys	Set to false if you want custom keys to be case-sensitive
ExpectedKeysFilePaths	The paths to json files containing the default Expected Keys
IgnoreCaseExpectedKeys	Set to false if you want expected keys to be case-sensitive

All other fields in **Properties.xml** are default values for the `Settings` field, and can be overwritten by changing these values in code.

3 Standard OCR Module

3.1 Overview

The Standard OCR module gives developers access to our Standard OCR engine.
This OCR engine has the following features:

- OCR from bit map, PDF or TIFF files or file streams
- OCR in memory bit maps
- Image Pre-processing and Auto-rotation
- Support for 23 languages
- .NET Programmatic and Zonal access to OCR results
- PDF Merging

3.2 Folders

The Standard OCR SDK contains the following folders:

- bin – Contains the binaries used by the Standard OCR module plus the required fonts and resources
- samples – contains samples (in C# and VB.NET) illustrating how to make use of the Standard OCR module and Barcode Module in common use cases

3.3 Application Development and Deployment

3.3.1 References

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

The Standard OCR bin folder, found at "[Install location]\Aquaforest SDK\OCR\Standard\bin", must be set as the resource folder when instantiating the OCR class. You must also have the files **"DPDFRenderNative_x86.dll**" (32-bit) and/or **"DPDFRenderNative_x64.dll**" (64-bit) in your output file path location. Without these files, the program will fail when attempting to use the OCR engine. These files can be copied from the Standard OCR bin folder.

3.3.2 Licensing

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

For example:

```
string license = "<your-license-key>";
Ocr ocr = new Ocr(license, resourcePath, logger);
```

3.4 Standard OCR API

The "samples" folder includes a number of sample application in C#, VB.NET and ASP.NET that make use of the Standard OCR API. The solutions provided are all created using Visual Studio 2017 and conversion to Visual Studio 2017 and above is handled automatically by that IDE.

3.4.1 PreProcessor Class

A `PreProcessor` object, which must be created and passed to the `Ocr` object, controls all of the preprocessing that can be performed on the input image in order to improve the quality of the output. Instantiation of the `PreProcessor` object will initialize a default set of pre-processing options which result in minimal image manipulation.

Constructor

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

Properties

Refer to the 'Aquaforest.Ocr' section of the "**Aquaforest SDK 3.0.chm**" file found in the folder "[SDK installation path]\docs\help" to view all the properties and methods available for the Standard OCR module.

3.4.2 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.

Constructor

```
Ocr ocr = new Ocr(license, resourcePath, logger);
```

or

```
Ocr ocr = new Ocr(license, resourcePath);
```

Properties

Refer to the 'Aquaforest.Ocr' section of the "**Aquaforest SDK 3.0.chm**" file found in the folder "[SDK installation path]\docs\help" to view all the properties and methods available for the Standard OCR module.

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

For C#, you can subscribe to this event through the following code:

```
ocr.StatusUpdate += OcrStatusUpdate;
...
private static void OcrStatusUpdate(object sender, StatusUpdateEventArgs pageCompletedEventArgs)
{
...
}
```

For VB, you can subscribe to this event through the following code:

```
Private Sub OcrPageCompleted(ByVal sender As Object, By Val statusUpdateEventArgs As
StatusUpdateEventArgs) Handles ocr.StatusUpdate
...
End Sub
```

3.4.3 StatusUpdateEventArgs Class

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

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

Properties

Property	Type	Description
BlankPage	bool	Indicates whether the page was detected as blank.
Confidence Score	double	Gets the confidence score. Generally a value of 1 or greater would indicate reasonable OCR of a page, but this should be confirmed by testing with "typical" source files.
ImageAvailable	bool	Indicates whether an image was successfully extracted (after applying all the appropriate pre-processing settings).
PageNumber	int	Returns page for which the object relates to.
Rotation	int	The rotation in Degrees (°) of the current page. If Autorotate is set to false this will always be 0.
TextAvailable	bool	Indicates whether text was extracted for the page.
Sc1 – Sc5	int	Advanced Setting – Refer to chm guide

3.4.4 Words Class

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

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.

Properties

Property	Type	Description
Count	int	Returns the number of WordData objects in a collection
Height	int	Returns the height of the current word
Width	int	Returns the width of the current word

Methods

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

3.4.5 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.

Properties

Property	Type	Description
AverageCharacterHeight	float	Gets the average height of a character
AverageCharacterWidth	float	Gets the average width of a character
Bottom	int	Gets the Y-coordinate of the bottom edge of the word in pixels
CharacterList	List<CharacterData>	Gets the list of characters in the word
Height	int	Gets the height of the word in pixels
Left	int	Gets the X-coordinate of the left edge of the word in pixels
Top	int	Gets the Y-coordinate of the top edge of the word in pixels
Width	int	Returns the width of the current word
Word	string	Gets the string representation of the word

3.4.6 PdfMerger Class

This class can be used to merge two PDFs.

Constructor

```
PdfMerger merger = new PdfMerger(sourceFilePath);
```

Methods

Property	Return Type	Description
Append(string pdfFileToAdd)	void	Appends the document specified to the PDF document in memory
Close()	void	Writes the output to the file specified in the constructor
Dispose()	void	Clears any resources not yet released. This is useful if an error occurs and you do not wish to write the merged output using Close()

3.4.7 Disposal and Temporary Files folders

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

3.4.8 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 behavior will result.

3.4.9 Words Class

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

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.

3.4.10 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 Aquaforest OCR SDK 2.30 Reference Guide Page 16

Language – The acceptable values are as follows:

0 - English
1 - German
2 - French
3 - Russian
4 - Swedish
5 - Spanish
6 - Italian
7 - Russian English
8 - Ukrainian
9 - Serbian
10 - Croatian
11 - Polish

12 - Danish
13 - Portuguese
14 - Dutch
19 - Czech
20 - Roman
21 - Hungarian
22 - Bulgarian
23 - Slovenian
24 - Latvian
25 - Lithuanian
26 - Estonian
27 – Turkish

MaxDeskew – Maximum angle by which a page will be deskewed

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

d2.2 – 2x2 dilation applied to all black pixel areas, useful for faint prints.

e2.2 – 2x2 erosion applied to all black pixel areas, useful for heavy prints.

c2.2 – closing process that performs a 2x2 dilation followed by a 2x2 erosion with the result that holes and gaps in the characters are filled.

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

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

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

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

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

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

4 Extended OCR Module

4.1 Overview

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

- IRIS OCR engine providing enhanced recognition
- Support for 138 languages. See section 4.5 for more details
- 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
- Optional Intelligent High-Quality Compression

4.2 Folders

The Extended OCR SDK ("[SDK installation path]\OCR\Extended) contains the following sub-folders:

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

4.3 Application Development and Deployment

4.3.1 References

A reference to `Aquaforest.ExtendedOCR.API` should be included in your application. You can also add a reference to `Aquaforest.ExtendedOcr.Shared` to access the results of the OCR on a word by word basis, allowing access to both word and character results, including positional information.

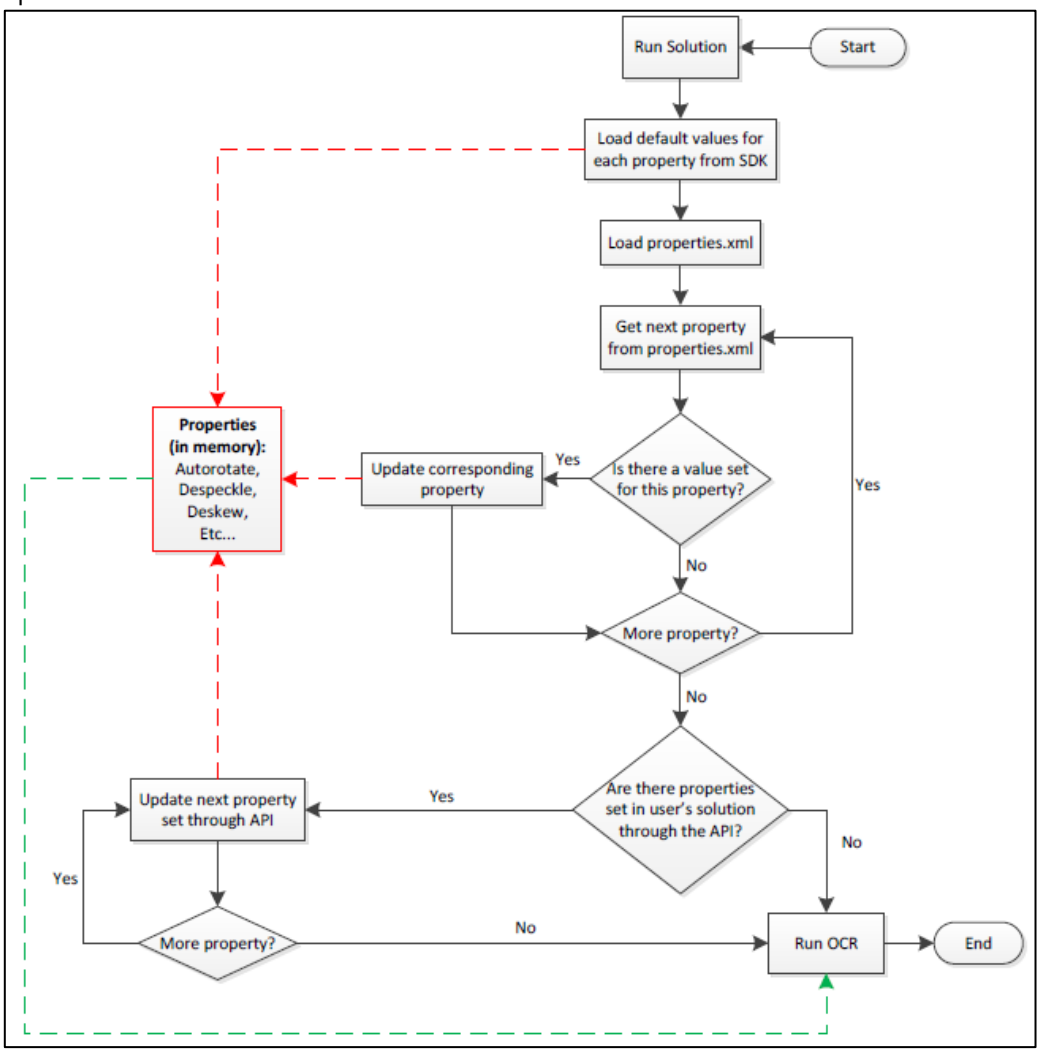
The Extended OCR bin folder, found at "[Install location]\Aquaforest SDK\OCR\Extended\bin\resources", must be set as the resource folder when setting up the Extended OCR Engine. You must also have the files **"DPDFRenderNative_x86.dll**" (32-bit) and/or **"DPDFRenderNative_x64.dll**" (64-bit) in your output file path location. Without these files, the program will fail when attempting to use the OCR engine. These files can be copied from the Extended OCR bin folder.

4.3.2 Properties.xml

The **Properties.xml** file located at "[SDK installation path]\OCR\Extended\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.

This is depicted in the flowchart below:



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

4.4 Extended OCR API

4.4.1 Classes

There are two classes used for the Extended OCR:

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

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

- `Words` – This class contains a collection of words, which contain all the data available for the words and characters for any given page.
- `WordData` – This class contains a collection of characters that make up a 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 this page.

4.4.2 Methods and Properties

Refer to the 'Aquaforest.ExtendedOcr' section of the "**Aquaforest SDK 3.0.chm**" file found in the folder "[SDK installation path]\docs\help" to view all the properties and methods available for the Extended OCR module.

4.4.3 Events

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

You can subscribe to this event through the following code:

```
ocr.StatusUpdate += OcrStatusUpdate;

[...]

private static void OcrStatusUpdate(object sender, StatusUpdateEventArgs
pageCompletedEventArgs)

{

    [...]

}
```

4.4.3.1 StatusUpdateEventArgs Class

Refer to the 'Aquaforest.ExtendedOcr' section of the "**Aquaforest SDK 3.0.chm**" file found in the folder "[SDK installation path]\docs\help" to view all the properties and methods available for the Extended OCR module.

Properties

Property	Type	Description
BlankPage	bool	Indicates whether the page was detected as blank
ImageAvailable	bool	Indicates whether an image was successfully extracted (after applying all the appropriate pre-processing settings).
PageNumber	int	Returns page for which the object relates to.
Resolution	int	The resolution of the page (DPI).
Width	int	The width of the page in pixels.
Height	int	The height of the page in pixels.
Rotation	int	The rotation in Degrees (°) of the current page. If Autorotate is set to false this will always be 0.
TextAvailable	bool	Indicates whether text was extracted for the page.
Pagelines	List<LineData>	Gets a list of recognized lines of text for the page.
Confidence Score	byte?	The average confidence score of the OCR for the page. The confidence score ranges from 0 (best confidence) to 255 (worst confidence). Note: Ocr.GetAdvancedOCRData must be set to 'True' to get the confidence score.
DetectedLanguages	List<LanguageCandidate>	Returns a list of detected languages when Ocr.LanguageDetection is enabled.

4.4.3.2 Words Class

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

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.

Properties

Property	Type	Description
Count	int	Returns the number of WordData objects in a collection
Height	int	Returns the height of the current word
Width	int	Returns the width of the current word
Item	WordData	Gets or sets the element at a specified index

Methods

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

4.4.3.3 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.

Properties

Property	Type	Description
Bottom	int	Gets the Y-coordinate of the bottom edge of the word in pixels
CharacterList	List<CharacterData>	Gets the list of characters in the word
Height	int	Gets the height of the word in pixels
Left	int	Gets the X-coordinate of the left edge of the word in pixels
Top	int	Gets the Y-coordinate of the top edge of the word in pixels
Width	int	Returns the width of the current word
Word	string	Gets the string representation of the word
AverageCharacterHeight	float	Gets the average height of a character
AverageCharacterWidth	float	Gets the average width of a character

4.4.3.4 CharacterData Class

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

Properties

Property	Type	Description
Baseline	int	Gets the Y-coordinate of the bottom edge of the character in pixels.
Character	string	Gets the string representation of the word
Height	int	Gets the height of the word in pixels
Width	int	Gets the width of the current word
X / Left	int	Gets the X-coordinate of the left edge of the character in pixels
Y / Top	int	Gets the Y-coordinate of the left edge of the character in pixels
Bottom	int	Gets the X-coordinate of the left edge of the word in pixels
Right	int	Gets the X-coordinate of the right edge of the word in pixels
AdvancedCharacterSettings	-	This property contains advanced information about the blob Note: 'Ocr.GetAdvancedOCRData' must be set to `true` to access this property

Advanced Settings

Property	Type	Description
ConfidenceScore	byte	The confidence score of the recognised blob (character) as computed by the OCR engine. The confidence score ranges from 0 (best confidence) to 255 (worst confidence).
IsBold	bool	True if the character is bold
IsItalic	bool	True if the character is italic
IsSubscript	bool	True if the character is subscript
IsSuperscript	bool	True if the character is superscript
ForegroundColor	Color	Gets the foreground color
BackgroundColor	Color	Gets the background color

4.5 Supported Languages

Extended OCR accepts up to 8 recognition languages at a time. This is helpful to process mixed documents but, because of the various character sets, not all combinations are allowed.

Multiple language support is limited to a single alphabet (or single alphabet plus English):

So, Russian (Cyrillic) and French cannot be mixed, neither can Japanese and Arabic.

4.5.1 Language 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	-
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	-
Korean	48	-

Name	Code	Description
Schinese	49	-Simplified Chinese
Tchinese	50	-Traditional Chinese
Quecha	51	-
Aymara	52	-
Faroese	53	-
Friulian	54	-
Greenlandic	55	-
Haitian_Creole	56	-
Rhaeto_Roman	57	-
Sardinian	58	-
Kurdish	59	-
Cebuano	60	-
Bemba	61	-
Chamorro	62	-
Fijan	63	-
Ganda	64	-
Hani	65	-
Ido	66	-
Interlingua	67	-
Kicongo	68	-
Kinyarwanda	69	-
Malagasy	70	-
Maori	71	-
Mayan	72	-
Minangkabau	73	-
Nahuatl	74	-
Nyanja	75	-
Rundi	76	-
Samoan	77	-
Shona	78	-
Somali	79	-
Sotho	80	-
Sundanese	81	-
Tahitian	82	-
Tonga	83	-
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	-

Name	Code	Description
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	-
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
Vietnamese	137	-
Thai	138	-

4.6 Fonts

Many of the common fonts that are used when reading PDF documents are already available. However, there is always a chance that the font that you need is missing. In this case, the `Recognize()` method will fail, and you will receive a message like below:

```
Process page 1 failed. Could not add '?' to the PDF.
Installing the fonts below can fix the issue.
mingliu.ttc

Alternatively, add them or any other fonts that support the language of the document
in the custom fonts folder 'D:\dev\source control\SDK-3.0\src\OCR\Cloud\bin\fonts'
```

Highlighted is the missing font, which should be available to download online. Once you download the font, you have two options:

Install Font – To install the font, you must make sure it has been unzipped, as zipped files cannot be installed. Right click on the font file and select "Install".

Add Font File to Font location – The font file can be moved inside the "bin" folder of CloudOCR, in a new folder called "fonts" Create this if it does not exist, and put the font file inside. The full path should be: *"[Install location]\src\OCR\Extended\bin\fonts\[font file]"*.

After following one of these steps, future runs of the program should be able to successfully use the newly added font.

5 Cloud OCR Module

5.1 Overview

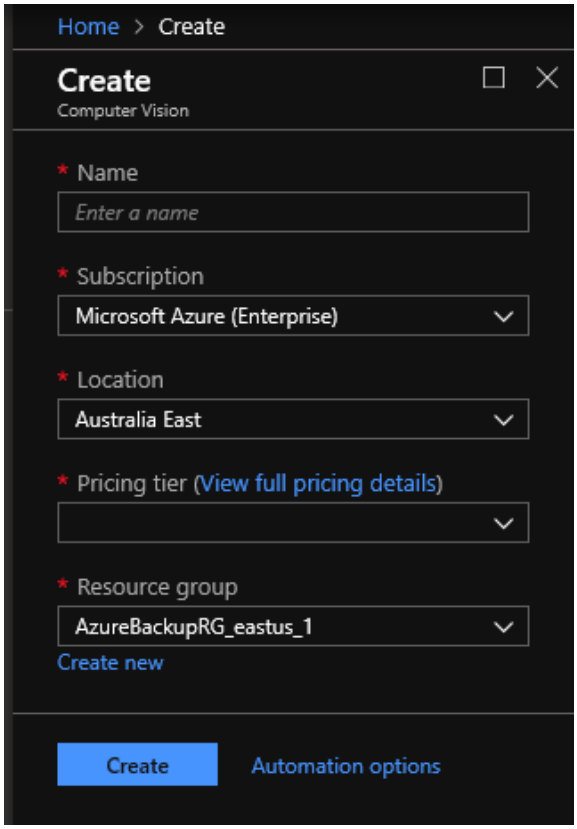
The Cloud OCR module extends the SDK by giving developers access to additional Google and Microsoft OCR engines. The advantage to using these OCR engines is their handwriting recognition capabilities and the range of OCR languages that can be read. These OCR engines are available as a SAAS model provided by both vendors. Before you start using these engines in your code, you must have a subscription to their service.

5.1.1 Microsoft Computer Vision

Azure's Computer Vision service provides developers with access to advanced algorithms that process images and return information. The image processing algorithms can analyze content in several different ways, depending on the visual features you are interested in. Computer Vision provides several services that recognize printed or handwritten text that appears in images.

To use this service, you will need a:

- Microsoft Azure account, you can sign up for this using the following link
- Microsoft Computer Vision API endpoint, you can add this to the azure account you created using the following link
    - Enter a suitable name for the endpoint
    - Choose your preferred azure subscription
    - Choose any location (Using a location that is closer to your files should give better performance)
    - Select a suitable pricing tier depending on your workload
    - Select or create a new resource group



Pricing

You can visit Microsoft's Pricing page using the following link. Select Offer, Region and Currency to view the pricing.

5.1.2 Google Cloud Vision

Cloud Vision API allows developers to easily integrate vision detection features within applications, including image labeling, face, and landmark detection, optical character recognition (OCR), and tagging of explicit content. We only use the OCR and Handwriting recognition features in the OCR SDK Cloud module.

To use the Cloud Vision API in the OCR SDK, you will need a:

- Google account, you can sign up for one using the following link
- Subscription key for the Google Cloud Platform. You can start your free trial using the following link, register for the trial and download your subscription key as a JSON file. Use the location of this JSON file as the value for the 'keyFilePath' when instantiating a new GoogleCloudOCR instance.

Pricing

You can visit Google's Pricing Calculator page using the following link, selecting "Cloud Vision" as the product, and putting your estimated monthly API calls in the "OCR" field. It allows you to accurately estimate the cost based on your monthly transactions, and even email or save these estimates to return to later. Note that you will consume one transaction per page processed.

5.2 Folders

The Aquaforest Cloud OCR Module contains the following folders:

- bin – Contains the binaries used by the Cloud OCR module
- samples – contains samples (in C#) illustrating how to make use of the Cloud OCR module, using both the Microsoft and Google engine

5.3 Application Development and Deployment

5.3.1 References

To use the Cloud OCR API, a reference to both `Aquaforest.CloudOCR.API` and `Aquaforest.CloudOCR.Shared` must be included. Your program should then have access to the Aquaforest API and both Cloud Engines.

The Cloud OCR bin folder, found at "[Install location]\Aquaforest SDK\OCR\Cloud\bin", must be set as the resource folder when instantiating the CloudOCR class. You must also have the files **"grpc_csharp_ext.x86.dll**" (32-bit) and/or **"grpc_csharp_ext.x64.dll**" (64-bit) in your output file path location. Without these files, the program will fail when attempting to use the Google Cloud engine. These files can be copied from the Cloud OCR bin folder.

5.3.2 Licensing

Production system deployment requires that a license string is defined in the code. The license string decides the number of concurrent processes that can be run. For Cloud OCR, you pass the license key to the constructor.

For example:

```
string license = "<your-license-key>";
CloudOcr ocr = new CloudOcr(license, resourcePath, logger);
```

5.3.3 Classes

There are four classes used in Cloud OCR. In addition to the `PreProcessor` and `CloudOcr` classes, there is `MicrosoftCloudOcr` specifically for the Microsoft OCR engine and `GoogleCloudOcr` specifically for the Google OCR engine:

- `PreProcessor` – This class configures and performs image pre-processing (such as de-skewing images) to ensure optimal OCR performance.
- `CloudOcr` – This is the class that configures and performs the Optical Character Recognition using the selected Cloud engine.
- `MicrosoftCloudOcr` – This is the class allows you to set the Microsoft OCR engine settings.
- `GoogleCloudOcr` – This is the class allows you to set the Google OCR engine settings.

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

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

5.3.4 Engine Settings

To perform OCR using `CloudOcr` class, you must first assign an engine to a `CloudOcr` instance. To set up each engine, you must instantiate them with the associated license information.

For example:

```
MicrosoftCloudOcr engine = new MicrosoftCloudOcr(endpoint, licensekey);
```

OR

```
GoogleCloudOcr engine = new GoogleCloudOcr(keyFilePath);
```

THEN

```
ocr.SetOcrEngine(engine);
```

5.3.4.1 MicrosoftCloudOCR Properties

Property	Description
TextRecognitionMode	Decide the type of text to recognize • `TextRecognitionMode.Printed` • `TextRecognitionMode.Handwritten`
HandwrittenRetries	The number of times to wait for the handwritten OCR results
HandwrittenWait	The amount of time (in milliseconds) to wait between each retry

Language	Select the language to use for OCR processing. This will determine the dictionary that is used. Auto-Detect will automatically detect the language for each page. 0 – Auto-Detect (default) 1 – Chinese (simplified) 2 – Chinese (traditional) 3 – Czech 4 – Danish 5 – Dutch 6 – English 7 – Finnish 8 – French 9 – German 10 – Greek 11 – Hungarian 12 – Italian 13 – Japanese 14 – Korean 15 – Norwegian 16 – Polish 17 – Portuguese 18 – Russian 19 – Spanish 20 – Swedish 21 – Turkish 22 – Arabic 23 – Romanian 24 – Serbian Cyrillic 25 – Serbian Latin 26 – Slovak

5.3.4.2 GoogleCloudOcr Properties

Property	Description
Languages	This property is a HashSet, so you can add multiple languages to use for OCR processing. This will determine the dictionaries that are used. Auto-Detect will automatically detect the language for each page and is the recommended one to use. 0 – Auto-Detect (default/recommended) 1 – Afrikaans 2 – Albanian 3 – Arabic 4 – Armenian 5 – Belorussian 6 – Bengali 7 – Bulgarian 8 – Catalan 9 – Chinese 10 – Croatian 11 – Czech 12 – Danish 13 – Dutch 14 – English 49 – SerbianLatin 50 – Slovak 51 – Slovenian 52 – Spanish 53 – Swedish 54 – Tamil 55 – Telugu 56 – Thai 57 – Turkish 58 – Ukranian 59 – Vietnamese 60 – Yiddish 61 – Amharic 62 – AncientGreek 63 – Assamese

	15 –Estonian 16 – Filipino 17 – Finnish 18 – French 19 – German 20 – Greek 21 – Gujarati 22 – Hebrew 23 – Hindi 24 – Hungarian 25 – Icelandic 26 – Indonesian 27 – Italian 28 – Japanese 29 – Kannada 30 – Khmer 31 – Korean 32 – Lao 33 – Latvian 34 – Lithuanian 35 – Macedonian 36 – Malay 37 – Malayalam 38 – Marathi 39 – Nepali 40 – Norwegian 41 – Persian 42 – Polish 43 – Portuguese 44 – Punjabi 45 – Romanian 46 – Russian 47 – RussianOldSpelling 48 – SerbianCyrillicAndLatin	64 – AzerbaijaniLatin 65 – AzerbaijaniCyrillic 66 – Basque 67 – Bosnian 68 – Burmese 69 – Cebuano 70 – Cherokee 71 – Dhivehi 72 – Dzonkha 73 – Esperanto 74 – Galician 75 – Georgian 76 – HaitianCreole 77 – Irish 78 – Javanese 79 – Kazakh 80 – Kirghiz 81 – Latin 82 – Maltese 83 – Mongolian 84 – Oriya 85 – Pashto 86 – Sanskrit 87 – Sinhala 88 – Swahili 89 – Syriac 90 – Tibetan 91 – Tigirinya 92 – Urdu 93 – UzbekLatin 94 – UzbekCyrillic 95 – Welsh 96 – Zulu

5.3.4.3 CloudOcr Properties

The OCR engines share most of their properties, so they have been built into the `CloudOcr` class. You can find information on these properties in the chm file, located at "[SDK installation path]/docs/help/Aquaforest SDK 3.0.chm".

5.4 Image Requirements

5.4.1 Microsoft Image Requirements

Supported Image Formats

- JPEG
- PNG
- GIF
- BMP
- PDF*
- TIFF*

File Size Limit – Less than 4MB

Image Dimensions – Between 50 x 50 and 4200 x 4200 (Image cannot be larger than 10 megapixels)

*These file types are converted to other filetypes during processing, so may fail at high resolutions

5.4.2 Google Image Requirements

Supported Image Formats

- JPEG
- PNG8
- PNG24
- GIF
- Animated GIF (first frame only)
- BMP
- WEBP
- RAW
- ICO
- PDF
- TIFF

File Size Limit – Less than 10MB

Image Dimensions – 1024 x 768 (recommended)

5.5 Fonts

Many of the common fonts that are used when reading PDF documents are already available. However, there is always a chance that the font that you need is missing. In this case, the `Recognize()` method will fail, and you will receive a message like below:

```
Process page 1 failed. Could not add '?' to the PDF.
Installing the fonts below can fix the issue.
mingliu.ttc

Alternatively, add them or any other fonts that support the language of the document
in the custom fonts folder 'D:\dev\source control\SDK-3.0\src\OCR\Cloud\bin\fonts'
```

Highlighted is the missing font, which should be available to download online. Once you download the font, you have two options:

Install Font – To install the font, you must make sure it has been unzipped, as zipped files cannot be installed. Right click on the font file and select "Install".

Add Font File to Font location – The font file can be moved inside the "bin" folder of CloudOCR, in a new folder called "fonts" Create this if it does not exist, and put the font file inside. The full path should be: *"[Install location]\src\OCR\Cloud\bin\fonts\[font file]"*.

After following one of these steps, future runs of the program should be able to successfully use the newly added font.

6 PDF Toolkit Module

6.1 Overview

The Aquaforest PDF Toolkit provides the developer the capabilities to:

- Create PDF Documents
- Convert CVS files to PDF
- Extract Text from PDFs
- Extract Images from PDFs
- Convert Images to PDF
- Merge PDFs
- Convert Office documents to PDF
- Convert PDF to PDF/A compliant files and validate
- Add Annotations to PDF pages
- Add/Extract Attachments from PDFs
- Add/Extract Bookmarks
- Use the PDF Toolkit Command Line Tool
- Get/Set PDF Metadata
- Get/Set PDF Viewer options
- Get/Set PDF Security
- Get/Set XMP Metadata
- Split PDFs
- Apply Stamps
- Add Overlays

In addition, the PDF Toolkit samples includes a customizable Command Line Toolset that provides an array of tools for creating, processing, and manipulating PDF files via the command line.

6.2 Folders

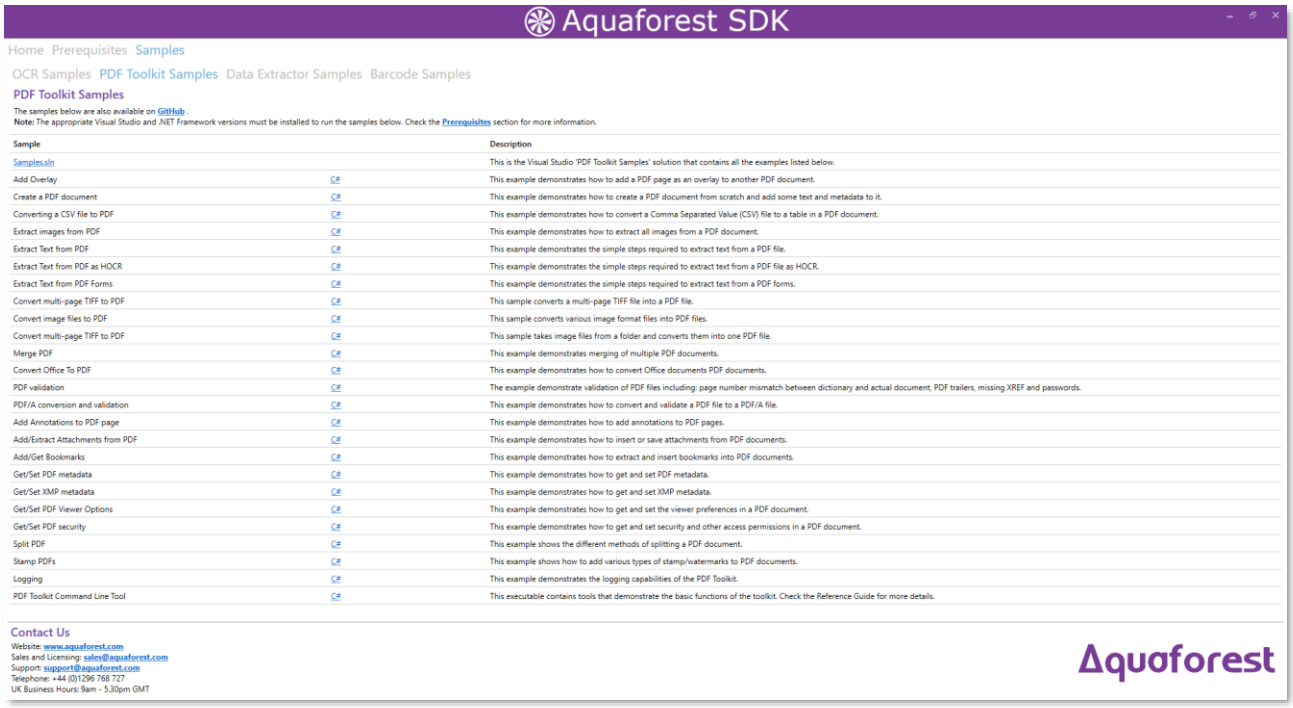
The Aquaforest PDF Toolkit Module contains the following folders:

- bin – Contains the binaries used by the PDF Toolkit module
- samples – contains samples (in C# and VB.NET) illustrating how to make use of the PDF Toolkit module

6.3 PDF Toolkit API

6.3.1 API Samples

The PDF Toolkit Samples tab contains links and explanation of all the sample projects that are shipped out with the PDF Toolkit. This includes an extensive Command Line Toolset project.



6.3.2 Product License Key

You will need to use the following method to apply your PDF Toolkit License key. This must be set before using any PDFToolkit tools.

```
PDFToolkit.LicenseKey = license;
```

6.3.3 API Documentation

The API documentation is available as a chm help file, located at "[SDK installation path]/docs/help/Aquaforest SDK 3.0.chm". It contains the full description of all the classes, methods and properties.

6.3.4 Deployment

A reference to `Aquaforest.PDF` will give you most of the functionality for the PDF Toolkit module in your program. There are a few situational API references that can be made in addition, such as the `Aquaforest.Office.PDF` reference for office conversions. Other references include `Aquaforest.PDF.Font` and `Aquaforest.PDF.Model`, and examples of their uses can be found in the sample projects.

6.3.4.1 C# and VB Deployment

Any deployment method should ensure that the target system meets the requirements (see section 1.2), and that the full contents of the PDF Toolkit bin folder is available.

7 Barcode Recognition 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 Folders

The PDF Toolkit module contains the following folders:

- bin – Contains the binaries used by the PDF Toolkit module
- samples – contains samples (in C# and VB.NET) illustrating how to make use of the PDF Toolkit module in common use cases

7.3 Application Development and Deployment

7.3.1 References

References to both "`Aquaforest.BarcodeReader.API`" and "`Aquaforest.BarcodeReader.Shared`" must be included in your program to be able to use the Barcode engine.

7.4 Barcode Module API

Refer to the "**Aquaforest SDK 3.0.chm**" file found in the folder "[SDK installation path]\docs\help" to view all the properties and methods available for the Barcode module.

8 Process Logging

The Aquaforest SDK uses a common logging object to allow process information to be logged. SDK modules use the logging object to return information to the process. It can also be used within your application.

There are two types of logging (plus no logging) and the detail level of the information logged can be specified.

8.1 Logging Types

There are four logging options provided in the Aquaforest SDK:

- Null – no logging
- SimpleConsoleLogger
- SimpleFileLogger
- Custom logger

If you do not need any logging, the logging object can be set to null (this is also the default if the parameter is not supplied).

8.1.1 SimpleConsoleLogger

The `SimpleConsoleLogger` writes information to the System Console. This console logger is thread safe.

8.1.2 SimpleFileLogger

The `SimpleFileLogger` writes logging information to a specified file, appending data to the file. There is an option to also write that information to the System Console. The file logger is thread-safe, application instances must write to their own log file.

8.1.3 Custom Logger

You can also create a custom logger by implementing the `IAquaforestLogger` interface (as well as with 3rd party logging providers) to log to whichever output you need.

8.2 Logging Levels

The logging level for a logging object is set during instantiation. You set the level as you require and only information with that logging level or higher is logged.

Log Level: Debug

This level contains detailed information on the process that is not normally of interest during the normal operation of the API, but might be of use while debugging your application.

Log Level: Information

This contains the type of information that a user would find of interest where the application is running interactively. It shows how the process is progressing.

Log Level: Warning

Within the SDK, this level will log warnings generated by the process that indicate that there is a potential problem that it does not consider to be an error.

Log Level: Error

Within the SDK, this level will log messages where the process has encountered a problem.

Log Level: Fatal

Within the SDK, this level will log messages where the process has failed and may possibly close.

Log Level: None

Setting the log level to none will stop all logging.

8.3 Usage

The simple console logger has the log level as an optional parameter (it defaults to Information).

```
IAquaforestLogger logger = new SimpleConsoleLogger(AquaforestLogLevel.Debug);
```

The simple file logger requires the output filename and has optional settings to output to Console and to set the log level.

```
IAquaforestLogger logger = new SimpleFileLogger(logFile, true, AquaforestLogLevel.Debug);
```

9 Acknowledgements

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

Name	Homepage
BitMiracle.LibTiff.NET	Homepage | GitHub
BouncyCastle.Crypto	Homepage
Camelot	Homepage
Cuneiform	n/a (Copyright (c) 1993-2008, Cognitive Technologies)
FreeImage.NET	Homepage
IKVM.NET	Homepage | Sourceforge
Leptonica	Homepage
Libjpeg	Homepage
Libpng	Homepage
Libtiff	Homepage
log4Net	Homepage
MahApps MahApps.Metro MahApps.Metro.IconPacks	Homepage GitHub GitHub
Newtonsoft.Json	Homepage | GitHub
PDFBox	Homepage
PDFMiner	GitHub
veraPDF	Homepage | GitHub
Zlib	Homepage
ZXing.NET	Homepage