Aquaforest OCR SDK for .NET Release Notes

Version 2.30
April 2019

Web: www.aquaforest.com
E-mail: info@aquaforest.com

Aquaforest

1 Version 2.30.190423

1.1 Bug Fixes

1.1.1 [SDK-120] Graphics state

The graphics state was not being restored when processing pages that require rotation in the Standard OCR engine. This caused issues when other applications manipulated the PDF after it had been OCRed by Aquaforest. This has now been fixed.

2 Version 2.30.190409

2.1 Changes

2.1.1 Visual C++ Redistributables

The SDK now only requires Visual C++ Redistributable 2017 for both OCR engines both for development and deployment.

2.2 Enhancements

2.2.1 New iDRS engine (Extended OCR module)

Aquaforest OCR SDK 2.30 now has the latest version of the iDRS engine (iDRS 15.4.5) in the Extended OCR module.

2.2.1.1 New languages available with High-Quality OCR engine

The brand-new technology 'High-Quality OCR' now embeds the 3 following languages:

- Italian
- Spanish
- Portuguese

Note also that variants of already existing High-Quality OCR languages are now supported as well: Afrikaans, Brazilian Portuguese, British, Corsican, Frisian, Luxembourgish, Mexican Spanish, Sardinian, and Swiss-German.

2.2.1.2 Performance improved for page orientation detection on Korean documents

The algorithm used for page orientation detection with Korean language has been reviewed, allowing to drastically reduce processing time while improving a bit the accuracy.

On a set of 132 Korean documents, taken in all possible orientations for a total of 528 test cases:

- Older versions:
  - Total time for orientation detection: 5,864 seconds
  - Orientation detection accuracy: 96,0%
- This version:
  - Total time for orientation detection: 971 seconds (divided by a factor 6!)
  - Orientation detection accuracy: 97,3%

2.2.1.3 Memory consumption reduced for document conversion

The document output engine includes several optimizations regarding memory consumption when creating an output document. Those changes impact mostly the creation of PDF Image-Text and especially PDF iHQC documents.

In terms of peak memory consumption, considering an input image A4 at 600DPI:

- Older versions:
  - PDF Image-Text: 343 Mb
  - PDF iHQC: 568 Mb
- This version:
  - PDF Image-Text: 238 Mb
  - PDF iHQC: 359 Mb

2.2.1.4 Compatibility between Arabic and Latin languages

This version can now be set up to recognize Arabic or Farsi languages with any Latin languages. The previous version only allowed Arabic (or Farsi) + English combinations.

2.2.2 [SDK-119] Confidence score

The Extended engine now provides access to confidence score at the page, word and character level. The confidence score ranges from 0 (best confidence) to 255 (worst confidence). `Ocr.GetAdvancedOCRData` must be set to `true` in order to get the confidence scores. The confidence score can be retrieved through the following method and properties:

- `Ocr.GetPageConfidence(pageNumber)`
- `WordData.ConfidenceScore`
- `CharacterData.AdvancedCharacterData.ConfidenceScore`

See the "ConfidenceScore" sample that comes with the SDK for more information. The sample is also available in the Cookbook and here.

2.2.3 New Properties

In addition to `ConfidenceScore`, several additional properties are now available in the Extended engine:

- `StatusUpdateEventArgs`
  - `PageLines`
  - `Resolution`
  - `Height`
  - `Width`

- `CharacterData.AdvancedCharacterData`
  - `IsBold`
  - `IsItalic`
  - `IsSubscript`
  - `IsSuperscript`
  - `ForegroundColor`
  - `BackgroundColor`

Consult the "chm" help file under "docs\help" in the installation folder for more information about these properties.

2.2.4 Hebrew Language

The Extended engine now has support for Hebrew language. You will need the Hebrew OCR license to use this feature.

2.2.5 Barcode blank page detection

The blank page detection in the barcode API is now performed for each pre-processing set defined. In previous versions, blank page detection was only done for the original un-preprocessed page(s). As a result, if a document was "blank" but had speckles in it, it would not be identified as blank even after applying pre-processing.

2.3 Bug Fixes

2.3.1 Processing file streams

When processing Stream inputs, the Aquaforest engine was throwing a `NullReferenceException` if `DebugOutput` was not set. This has been now been fixed

2.4 Known Issues

2.4.1 Recognition of accented characters with High-Quality OCR engine (Extended OCR module)

The new Extended OCR module currently has an issue that impacts Latin languages processed with High-Quality OCR engine.

When a character with an accent (like é, è, à, ñ, etc.) is recognized but is not present in the character set (for instance if recognition is performed in English), the OCR engine will output a reject character (U+FFFD). This is a regression compared to previous versions, where the "base" character would be output instead (e.g. 'e' instead of 'é').

This issue will be fixed with the next release of iDRS.

3 Version 2.30.190128

3.1 Changes

3.1.1 Visual C++ Redistributables

SDK 2.30 now requires Visual C++ 2010 and 2017 for both OCR engines both for development and deployment.

3.1.2 Default values

The default values for a few settings have been changed so that it gives good OCR results for different types of documents. These are shown below:

- Aquaforest OCR Module

Setting	Changed to
SavePreDespeckle	true

- Extended OCR Module

Setting	Changed to
Binarize	true
BinarizationMode	Adaptive
Brightness	128
SmoothingLevel	248
Threshold	0
WorkDepth	255
RemoveLines	true

3.2 Enhancements

3.2.1 New iDRS engine (Extended OCR module)

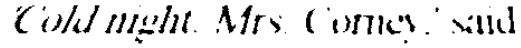
Aquaforest OCR SDK 2.30 now has the latest version of the iDRS engine (iDRS 15.4.3) in the Extended OCR module.

3.2.1.1 New High Quality OCR engine

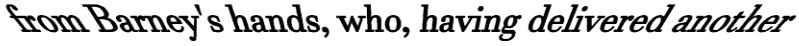
The iDRS™ is updated with I.R.I.S.' brand new High Quality OCR: a new OCR engine developed using state of the art concepts from the artificial intelligence research domain.
This new technology brings considerable OCR accuracy improvement especially for bad quality scans, camera images or low resolution documents, which are affected by common issues such as:

- Touching characters

- Broken characters



- Distorted characters



It will also be suited for recognition of Arabic and Farsi, due to the cursive nature of these languages:

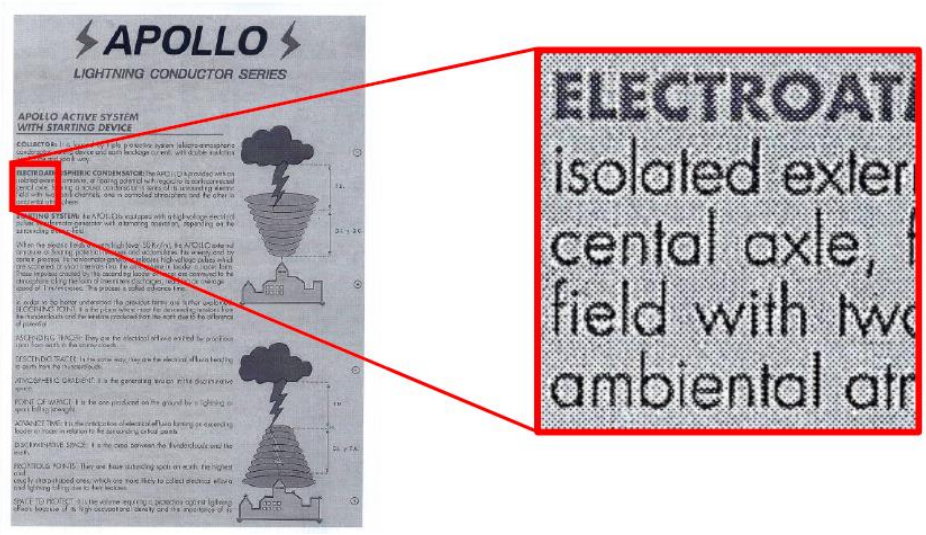
The first release uses High Quality OCR engine for English, Arabic and Farsi languages; further languages will of course be added in future releases.

- For Latin, Cyrillic, Greek, Hebrew and Asian languages, High Quality OCR will be combined with existing OCR engine in order to use the strengths of both engines.
- For Arabic and Farsi languages, it fully replaces the previous engine, and reaches an unparalleled level of accuracy.

Note that processing time with High Quality OCR engine is expected to increase for low-quality documents: more time will be spent but better accuracy will be reached.

3.2.1.2 Recognition of images scanned with dithering

This release expose an option allowing to improve recognition of colour or greyscale images scanned with dithering:



Previous releases would not have properly processed such images: in most cases, the text would simply not have been detected during page analysis step.

How to use

It can be enabled by setting the `Undithering` property in the `Binarization` object. Note that you also need to enable smoothing by setting `SmoothingLevel` to a value greater than '0' in order to perform undithering.

3.2.1.3 Automatic language detection of a single-language page

Extended OCR can now automatically detect the language of an input document.
The aim of this feature is to detect the most probable language of a single-language page.

Supported languages

This release will be able to reliably detect the following scripts/languages:

- **Latin script**
  English, German, French, Spanish, Italian, Swedish, Danish, Norwegian, Dutch, Portuguese, Galician, Icelandic, Czech, Hungarian, Polish, Romanian, Slovak, Croatian, Slovenian, Finnish, Turkish, Estonian, Lithuanian, Latvian, Albanian, Catalan, Irish Gaelic, Scottish Gaelic, Basque, Indonesian, Malay, Swahili, Tagalog, Haitian Creole, Kurdish, Cebuano, Ganda, Kinyarwanda, Malagasy, Maltese, Nyanja, Sotho, Sundanese, Welsh, Javanese, Azeri (Latin), Uzbek, Bosnian (Latin), Afrikaans
- **Cyrillic script**

Serbian, Russian, Byelorussian, Ukrainian, Macedonian, Bulgarian, Kazakh
- **Greek script**
  Greek
- **Hebrew script**
  Hebrew

Future releases will extend the support to Arabic and Asian scripts.

How to use
To enable this feature, set the `LanguageDetection` property of the `Ocr` class to `true`.

To get the language detection results of each page, use the `StatusUpadate` event handler and access the `DetectedLanguages` property. A list with 3 candidate languages and their respective confidence level will be provided. `SupportedLanguages.None` with confidence level of 255 is used to indicate that the detection could not succeed in finding a relevant candidate. For instance, the following results may be returned:

- *English - confidence level 10; French - confidence level 50; Italian - confidence level 80*
  Language detection could find 3 language candidates, English being the most probable and Italian the least.
- *English - confidence level 10; None - confidence level 255; None - confidence level 255*
  Language detection could find only one language candidate, English.
- *None - confidence level 255; None - confidence level 255; None - confidence level 255*
  Language detection could not find a relevant language candidate on the document.

Note:
- If at least one language has been detected, recognition will be performed in the first language candidate that has been detected, and not in the language(s) set through the `Language` or `Languages` property.
- If it fails to detect a language, recognition will be performed using the language(s) set through the `Language` or `Languages` property (same behaviour as before).

3.2.2 Punch-hole removal

A new feature has been added to the Extended engine that attempts to remove punch holes from pages. This feature only works when converting images to PDFs or when OCRing PDFs with `ExtractImageMethod` set to `ConvertToTiff` and with either `KeepOriginalImage` set to `false` or `KeepPunchHoleRemoval` set to `true`. Punch-hole removal can be enabled by setting `PreProcessor.RemovePunchHoles` to `true`.

Note: The punch-hole algorithm can be used on images with the following minimum dimensions width: 300px, height: 100px (computed for 300 DPI). The minimum height and width can vary with the image resolution.

3.2.3 Retain pre-processing settings

You can now retain specific pre-processing in the output PDF documents. For instance, if de-speckling is enabled, speckles are removed from each page to improve the OCR recognition but this is only done internally and are not reflected in the output PDF document.

In this release, if you want to retain the de-speckling in the output document, set `KeepDespeckle` to `true`. Other pre-processing settings that can be preserved are deskew, dark border removal and punch-hole

removal, which can be enabled using `KeepDeskew`, `KeepDarkBorderRemoval` and `KeepPunchHoleRemoval` respectively.

This feature only works when converting images to PDFs or when OCRing PDFs with `ExtractImageMethod` set to `ConvertToTiff` and with `KeepOriginalImage` set to `false`.

3.2.4 Advanced pre-processing settings

This release has new advanced settings for some existing pre-processing settings of the Extended module. These are:

- AdvancedDeskew
  - AdjustmentMode
  - ForceDeskew
- AdvancedDespeckle
  - Dilate

See the help file (*.chm) under "[sdk installation folder]\docs\help" for more information regarding these settings.

3.2.5 Turn off PDF/A validation

In previous versions, PDF/A validation was always performed after converting to PDF/A. However, validating a PDF/A document adds a small performance penalty in terms of the overall processing time. This version allows you to turn off PDF/A validation.

In the Aquaforest engine, this can be achieved by setting the `validatePDFA` parameter of `Ocr.SavePDFAOutput(...)` to `false`.

In the Extended engine, this can be achieved by setting `Ocr.ValidatePDFA` to `false`.

4 Version 2.20.170906

4.1 Bug Fixes

4.1.1 Cyrillic Languages

Fixed issue with OCRing documents with Cyrillic languages. The characters were not being encoded correctly in the output document.

5 Version 2.20.170707

5.1 Changes

5.1.1 Minimum PdfToImageDpi

The minimum PdfToImageDpi is now 72 instead of 150. This can be set either through the API or the Properties.xml.

5.2 Bug Fixes

5.2.1 PDF Forms

Fixed issue with processing PDF documents containing text or images inside forms.

5.2.2 OCRed Text Placement

In documents with very large pages, the OCR text placement was not very accurate. This issue has been fixed in this version.

6 Version 2.20.170410

6.1 Changes

6.1.1 .NET Framework

The Aquaforest engine has been upgraded to use .NET Framework 4.5.2 instead of .NET Framework 3.5. Consequently, there is no need now to specify "`useLegacyV2RuntimeActivationPolicy`" in the `app.config` file of your application.

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

6.1.2 VC++ Redistributables

This version requires VC++ 2013 instead of VC++ 2012 for the Aquaforest engine.

6.1.3 PDFBox

The SDK now uses PDFBox instead of iTextSharp for reading PDF documents.

6.1.4 PDF/A

To generate PDF/A files, "`ocr.EnabledPDFAOutput`" needs to be set instead of "`ocr.EnabledPdfOutput`".

If the source file is a PDF file, "`ocr.ConverToTiff`" needs to be set to `true` because conversion to PDF/A is no longer available in "Native" mode.

Additionally, the "`ConvertPDFToPDFA`" method which previously enabled creating PDF/A files without going through the OCR process, is no longer available as there were lots of instances where invalid PDF/A outputs were generated.

6.1.5 PdfToImageEngine

This version of the SDK only has 1 engine that converts PDF pages to images. All occurrences of "`PdfToImageEngine`" have been removed from the API and the Properties.xml.

6.1.6 Stamps

Stamps have been removed in this release. To use stamps, you need to use the PDF Toolkit, which is available to users with Advanced and Extended License.

6.2 Enhancements

6.2.1 Additional Language Support

The Extended engine now supports Arabic and Farsi languages.

6.2.2 Process PDF files with vector objects in native mode

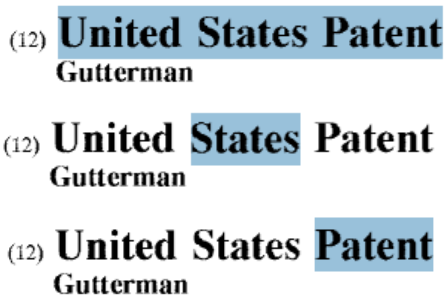
PDF documents that contain only vector images (e.g. CAD drawings) can now be OCRed natively. In previous versions, the PDF needed to be re-imaged (`ConvertToTiff`) before OCRing.

By default, pages that contain only vector objects are rasterized. Pages that do not have any images but contain vector objects as well as electronic text are skipped from rasterization. However, sometimes there can be a page that contains vector objects (CAD drawings) but its title may be in electronic text. To force

rasterizing pages like these, there is a property called "`PdfToImageForceVectorCheck`" which needs to be set to true. This property can be set through the `OCR` object or the Properties.xml file.

6.2.3 Font sizing

The sizing of OCRed text added to PDF documents has been improved in both OCR engines. This can be tested by selecting the OCRed text in a PDF reader.

Previous versions	New version
	

6.2.4 Retain Viewer Preferences

A new property (`ocr.RetainViewerPreferences`) has been added to enable retaining the PDF viewer preferences (Page Layout, Page Mode) when OCRing PDF and using "`ConvertToTiff`". The viewer preferences are automatically retained when processing in "Native" mode.

6.2.5 FIPS Compliancy

The SDK is now FIPS compliant.

6.2.6 Additional settings in the Extended engine

The following settings have been added to the Extended engine:

- BinarizationMode
- TextSpacing
- TextType
- CharactersPerInch
- LimitCharsetCharacters
- UserLexicon

Refer to the **OCRSDK 2.2 (Extended engine).chm** file found in the folder "[SDK installation path]\docs\help" to view description of these new settings.

7 Version 2.10

7.1 Changes

7.1.1 Licensing

There are a couple of 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		✓	✓	✓
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				✓

Function	Basic Edition	Standard Edition	Advanced Edition	Extended Edition
Intelligent High Quality Compression				✓

7.2 Enahancements

7.2.1 Barcode Decoding

SDK 2.10 now supports decoding barcodes from images and PDF documents. The following barcode formats are currently supported:

- 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.
- 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

7.2.2 New iDRS engine (Extended OCR Module)

Aquaforest OCR SDK 2.10 now has the latest version of the iDRS engine (iDRS 15) in the Extended OCR module. It provides the following new features:

- Improved character recognition engine
- Additional output formats:
    - PDF/A-1a
    - EPUB (short for electronic publication) is an e-book standard by the International Digital Publishing Forum (IDPF). Files have the extension .epub.
- New Asian OCR engine. It now has two Asian OCR engines to process documents with Asian languages (Japanese, Korean, Simplified Chinese and Traditional Chinese). The engine to use can be set using the `AsianOcrEngine` property in the `Ocr` class.

7.2.3 System.IO.Stream Input & Output

SDK 2.10 now provides the option of reading input source from `System.IO.Stream` and saving output results to `System.IO.Stream`.

7.2.4 Retain Bookmarks

Added support for retaining bookmarks in the output PDF document when OCRing a PDF source and using `ConvertToTiff`.

7.2.5 Retain Metadata

Added support for retaining metadata in the output PDF document when OCRing a PDF source and using `ConvertToTiff`.

7.2.6 SMask support

Added support for PDF files with SMask images.

7.2.7 LogFilePath

A new `LogFilePath` property has been added to enable users to get the location of the debug log file path when `EnableDebugOutput` is set to `DebugLevels.LOG_FILE_PATH`

7.3 Bug Fixes

7.3.1 Merging PDFs

Merging certain PDF files was causing the resulting merged PDF to be very large. This has now been fixed.

7.3.2 Hanging

The Aquaforest OCR SDK was hanging on certain machines when subjected to extreme load such as running in multi-core. This has now been fixed.