Aquaforest
Aquaforest SDK 3.2
Upgrade Guide

Aquaforest SDK
Upgrade Guide

Version 3.2
March 2024

Content

1 Introduction

If you are upgrading an existing solution from a previous Aquaforest development kit (PDF Toolkit or OCR SDK), there are a number of minor changes that you will need to make to your solutions.

To assist you in this process, this guide includes walk-throughs of upgrading samples from the OCR SDK 2.3 and the PDF Toolkit 2.0 to use the Aquaforest SDK 3.0 and above.

The walkthroughs are for

- Barcode Reader sample
- Standard OCR sample
- Extended OCR sample
- PDF Toolkit sample

In addition, most of the previous PDF Toolkit and OCR SDK samples have equivalent samples in the new SDK. These will provide suitable templates for your solutions.

Previous versions of the development kits will need to be uninstalled before Aquaforest SDK 3 can be installed. It might be worth copying the sample folders from the previous version's before uninstalling.

1.1 Licensing

Previously the Aquaforest PDF Toolkit and OCR SDK used different license keys. The new Aquaforest SDK uses a single license key across all the modules.

The Reference Guide has information about how the license key is applied to the different modules.

2 Upgrade Walkthroughs (OCR SDK 2.3 to SDK 3.0 and above)

The following upgrade walkthroughs make use of the Samples solution supplied with the OCR SDK 2.3 and upgrades them to use the Aquaforest SDK 3.0 and above.

It uses the original OCR SDK 2.3 sample code as supplied, your project may well have the license information already in place, though the license will need to be updated.

2.1 Barcode Reader

2.1.1 Update solution

- Open the solution.
- If the Solution Explorer is not visible, click on the View menu item and select Solution Explorer from the menu.
- For each project in the solution
  - Right click on each project and select Properties.
  - On the properties page, select Application.
  - Change the Target Framework to .Net Framework 4.7.2.
  - Save the change.

2.1.2 Update References

- Remove the existing barcode SDK references.
- Using File Explorer remove the existing bin folder for the project.
- Copy the contents of the bin folder from `<installation folder>\Barcode` to the projects Build Output folder.
- Add references to the following in the bin folder:
  - `Aquaforest.BarcodeReader.API`
  - `Aquaforest.BarcodeReader.Common`
  - `Aquaforest.Logging.`

2.1.3 Add license key

- Open the Helper.cs file and add your SDK license key and remove/comment out the #error.
- Save the file.

2.1.4 Code changes

There are some minor changes to the code

Select Resource folder

Replace the existing `string resourceFolder =...` code with

```
Assembly asm = Assembly.GetExecutingAssembly();
string resourceFolder = Path.GetDirectoryName(asm.Location);
```

At this point you will have one error:

```
'DecodeResult' does not contain a definition for 'BarcodeResult' and no accessible
extension method 'BarcodeResult' accepting a first argument of type 'DecodeResult'
```

`BarcodeResult r = result.Value.BarcodeResult;`

The new SDK has replaced the property BarcodeResult with a BarcodeResults collection property.

To obtain the results replace the above code with:

```
foreach (BarcodeResult r in result.Value.BarcodeResults)
  {
      Existing code...
  }
```

Wrapping it around the `Console.Write`.

When executed, it may list missing files (in the debug/bin folder bar the last).

If it lists missing files either add a reference to the file or add the file to bin folder from the SDK

Add

```
Console.WriteLine("Press any key");
Console.ReadKey();
```

To the Main() so you can see the results.

2.1.5 Adding Logging

Logging is easily implemented in the sample by creating either a SimpleConsoleLogger or a SimpleFileLogger instance and passing it to the Barcode reader.

BarcodeReader reader = new BarcodeReader(settings, logger);

See section 4 for more details.

2.2 Standard OCR

2.2.1 Update solution

Open the solution.

- For each project in the solution
  - Right click on each project and select Properties.
  - On the properties page, select Application.
  - Change the Target Framework to .Net Framework 4.7.2.
  - Save the change.

2.2.2 Update References

- Remove the existing Aquaforest/Standard OCR reference.
- Using File Explorer remove the existing bin folder for the project.
- Copy the contents of the bin folder from `<installation folder>\OCR\Standard` to the projects Build Output folder.
- Add references to the following in the bin folder:
  - Aquaforest.OCR.API
  - Aquaforest.Logging.

2.2.3 Add license key

- Open the Helper.cs file and add your SDK license key and remove/comment out the #error.
- Save the file.

2.2.4 Code changes

Define the location of the resource folder.

```
Assembly asm = Assembly.GetExecutingAssembly();
string resourceFolder = Path.GetDirectoryName(asm.Location);
```

Replace:

```
Using ocr As New Ocr()
```

With

```
using (Ocr ocr = new Ocr(Helper.LICENSE_KEY, resourceFolder))
```

Remove

```
ocr.License = Helper.LICENSE_KEY;
```

```
ocr.ResourceFolder = resourceFolder;
```

This is now part of the constructor.

You also need to remove:

```
        ocr.EnableConsoleOutput = true;
```

This is now handled by the Aquaforest Logger.

Sometimes you need to add a reference to NetStandard (only some samples). Follow the Quick Actions suggestions from the IDE.

2.2.5 Adding Logging

Logging is easily implemented in the sample by creating either a SimpleConsoleLogger or a SimpleFileLogger instance and passing it to the OCR constructor.

Add two "`using`":

```
using Aquaforest.Logging;
```

```
using Aquaforest.Logging.Abstractions;
```

Add the logger before the `using (Ocr ocr`…:

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

If you alternatively want to write the log entries to a file, use:

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

Replace

```
using (Ocr ocr = new Ocr(Helper.LICENSE_KEY, resourceFolder))
```

with:

```
using (Ocr ocr = new Ocr(Helper.LICENSE_KEY, resourceFolder, logger))
```

See section 4 and the Reference Guide for more details on the logging and logging levels.

2.3 Extended OCR

2.3.1 Update solution

Open solution.

- For each project in the solution
  - Right click on each project and select Properties.
  - On the properties page, select Application.
  - Change the Target Framework to .Net Framework 4.7.2.
  - Save the change.

2.3.2 Update References

- Remove existing Extended OCR references
- Remove the bin folders.
- Copy the bin folder from <installation folder>\OCR\Extended to the projects Build Output folder.
- Add references to the following in the bin folder:
  - Aquaforest.ExtendedOCR.API
  - Aquaforest.ExtendedOCR.Shared
  - Aquaforest.Logging.

2.3.3 Add license key

- Open the Helper.cs file and add your SDK 3.1 license key and remove/comment out the #error.
- Save the file.

2.3.4 Code changes

Define resourceFolder as:

```
 string resourceFolder = Path.GetFullPath(<Build Output Folder>\Resources);
```

Add two "`using`":

```
using Aquaforest.Logging;
```

```
using Aquaforest.Logging.Abstractions;
```

Sometimes you will also need to add a reference to NetStandards, just follow the Quick Actions prompt.

Replace:

```
using (Ocr ocr = new Ocr(resourceFolder))
```

With:

```
            ExtendedOcrEngine.SetupOcr(Helper.LICENSE_KEY, resourceFolder);

            IAquaforestLogger logger = new
SimpleConsoleLogger(AquaforestLogLevel.Information);

            using (Ocr ocr = new Ocr(logger))
```

Remove the following lines:

```
            ocr.License = Helper.LICENSE_KEY;

            ocr.EnableConsoleOutput = true;
```

As they are dealt with by the constructor and the Aquaforest logger.

2.3.5 Adding Logging

Logging has been added in the code changes above. See section 4 for more details.

2.3.6 Note:

Some of the SDK 2.3 Extended OCR sample projects, including DetermineWordCoordinates and GetTextFromPage require additional work. They use a static instance of the OCR object, which is difficult if not impossible to implement using the new (Extended) OCR engine.

Changes required:

Remove declaration of static Ocr object ocr.

Remove disposal of non-null ocr object.

Add:

```
                string resourceFolder = Path.GetFullPath(<Build Output
Folder>\Resources)

                ExtendedOcrEngine.SetupOcr(Helper.LICENSE_KEY, resourceFolder);

                IAquaforestLogger logger = new
SimpleConsoleLogger(AquaforestLogLevel.Information);

                using (Ocr ocr = new Ocr(logger))
```

3   Upgrade walkthrough (PDF Toolkit)

The following walkthrough uses the samples as supplied with the PDF Toolkit 2.0.

3.1.1   Update solution

Open solution.

- For each project in the solution
    - o   Right click on each project and select Properties.
    - o   On the properties page, select Application.
    - o   Change the Target Framework to .Net Framework 4.7.2.
    - o   Save the change.

3.1.2   Update References

- Remove existing Aquaforest.PDF references
- Remove the bin folders.
- Copy the bin folder from <installation folder>\PDFToolkit to the projects Build Output folder.
- Add references to the following in the bin folder:
    - o   Aquaforest.PDF
    - o   Aquaforest.Logging

Do not be alarmed if the error count remains high, the Visual Studio IDE does not update the "using" references automatically (even when you Rebuild), you need to visit each file in turn (just click on the Errors in the Error List to open the file – the action of opening the file updates the reference).

3.1.3   Add license key

The PDF Toolkit 2.0 samples do not make use of a Helper file, each project has its own license key assignment which will need to be updated.

```
PDFToolkit.LicenseKey = "<license key>";
```

The SDK 3.1 license key is much shorter than the previous PDF toolkit license key.

3.1.4   Code changes

Some additional "usings" are required for some of the sample projects

3.1.4.1   Project StampPDF

- Add `using Aquaforest.PDF.Model;`

3.1.4.2   Project CSVToPDF

In the file CSVToPDF.cs in the project CSVToPDF, the static enum PageSize will require the parent type prepended, in this case PDFPage:

```
pageSettings.SetSize(PDFPage.PageSize.A4);
```

3.1.4.3   StampPDF

The StampVariable method used in StampPDF.cs has changed its overload options with the addition of a Boolean flag indicating whether the stamp should be an image or text.

Original code:

```
stamper.StampVariable(variable, position, page);
```

Revised code:

```
bool stampAsImage = false;

stamper.StampVariable(variable, position, page,stampAsImage);
```

The StampPageNumberBates method has also changed its overload options. Dependent upon your requirements, you will have to change the signature which best meets your requirements.

In the sample file StampPDF.cs, there is a call to StampPageNumberBates.

```
stamper.StampPageNumberBates(7, suffix, prefix, 1, 1, position);
```

This can be replaced with:

```
bool stampAsImage = false;

stamper.StampPageNumberBates(7, suffix, prefix, 1, position, stampAsImage);
```

These changes will correct all the compilation errors.

3.1.5 Adding logging

See section 4 for more details.

4 Logging

Logging is fully integrated into the design of the modules of the new Aquaforest SDK.

Implementing the logging in your application is very simple, just requiring the creation of either a SimpleConsoleLogger or a SimpleFileLogger instance and passing it to the relevant module.

Add two "`using`":

```
using Aquaforest.Logging;

using Aquaforest.Logging.Abstractions;
```

4.1 Simple Console Logger

The Simple Console Logger displays output to the standard console (normally the display).

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

4.2 Simple File Logger

The Simple File Logger writes the output to a specified file, with an option to additionally write to the standard console.

```
        IAquaforestLogger logger = new SimpleFileLogger(logFile, writeToConsole,
AquaforestLogLevel.Information);
```

4.3 Logging Level

The AquaforestLogLevel sets the level of log entries recorded (the Debug level also sets internal settings to Debug, similar to the old EnableDebugOutput setting).

See Reference Guide for more details on the logging levels.