

Aquaforest OCR SDK – Getting Starting Video Tutorial Transcript

The Aquaforest OCR SDK for .Net incorporates the same high performance OCR engine that is included in the Aquaforest TIFF Junction and Autobahn DX products. The SDK API allows developers full control over OCR processing to enable customized integration of OCR within .Net applications.

Key features and benefits :

- The SDK can accept as input Bitmap, multi-page TIFF and PDF Files.
- It can create Searchable PDF, RTF or Text output files.
- Control pre-processing options such as despeckle, deskew, line removal and autorotate.
- Specify one of over 20 supported document languages.
- Enumerate the OCR results, examining the words and characters recognized along with their co-ordinates.
- Process multi-page TIFF and PDF files one page at a time or all in one operation

Samples

The samples folder includes a number of sample applications in C#, VB.Net and ASP.Net. The project files are for Visual Studio 2008. The sample applications are as described below.

Simple Console Application

This includes VB and C# version of the simple application listed in section 2 of the reference guide.

ConvertFolderOfFiles

This forms-based application demonstrates converting a folder of TIFF or Bitmap files. Both VB.Net and C# versions are included.

ZonalOCR

This forms-based application demonstrates converting a folder of TIFF or Bitmap files. Both VB.Net and C# versions are included.

ASP.Net

This demonstrates a simple web-based application that allows uploading an image file for OCR and conversion to text, RTF or searchable PDF.

Demonstration Application

In this tutorial we will create a simple console application in C# to convert a sample TIFF file to Searchable PDF. Here we are using Visual Studio 2008 but other versions such as 2005 and 2010 can also be used as long as version 3.5 of the .Net framework is available.

We first need to add a reference to the Aquaforest OCR SDK API DLL. This is in the SDK bin folder.

We will also configure the application to build as a 32-bit x86 app. This will be able to run on both 32-bit and 64-bit systems.

Objects

There are two objects that need to be created. The first is the OCR object which is used for most of the major operations within the SDK. The second is the pre-processor object which is used to configure pre-processor settings such as Deskew and Despeckle.

Resource Folder

The OCR SDK resource folder is the bin folder containing the DLLs, dictionaries and data files used by the OCR engine. The location of this needs to be set. In addition it is also necessary to add 3 native DLL files and a fonts folder to the bin folder of the application. Just copy the files from the bin_add folder in the SDK installation.

Enable Output

Next we need to enable PDF as an output type by setting enable pdf output to true. Also, for the purposes of the demo we will enable console output so that we can see messages as each page is being processed.

Read Source and Recognize

Now we read the source file – from the docs/tiffs folder in the SDK installation folder. Then we run the Recognize method to perform the OCR before finally saving the PDF output with the SavePDFOutput method.

Note that it is possible to run the OCR in a much more granular fashion being able to programmatically navigate the recognition results but for this demo we are just creating a searchable PDF.

Finish Up

Finally we clean up the temporary files used by the engine with the DeleteTemporaryFiles method.

Run

Now let's build and run. We can see the process messages as each page is processed. Once complete, we can go ahead and check the generated searchable PDF file.