User Friendly Consulting

White Paper: ABBYY Recognition Server Web Service API Example

By: Joe Hill

Published: June 2017

Summary

ABBYY Recognition Server converts paper or electronic documents into compressed, searchable, archive compliant files. It also provides the ability to extract text and barcodes from paper or electronic documents and return the results in XML format. This not only includes the textual content that was found but also more detailed information such as the paragraph, line locations and the text formatting. This information may be used to construct additional applications that perform operations such as document indexing and searching.

This document will show how the ABBYY Recognition Server web service API can be used to convert a document into an archival format. In addition, it will show how to extract text and barcodes from a document and how to use this information to power a document search application. This example is shown in the context of Microsoft Visual Studio.NET but the concepts apply to any development and runtime environment that can call a web service such as Java. Example VB.NET code will be presented for example purposes only.

About ABBYY Recognition Server Administration

ABBYY Recognition Server is administered using the Remote Administration Console. This tool provides the ability to configure the processing stations, workflows, users, and licensing. It also provides the ability to monitor jobs and to view the event log which shows informational messages as well as warning and error messages for the entire system. The rest of this document will assume that the ABBYY Recognition Server has been properly installed with a valid software license activated. Any warning or error messages in the event log should be resolved before moving forward with use of the API.

The basis of the administration of Recognition Server is done through the configuration of a workflow. Workflows provide specifications for document processing operations. Each specifies a document source, processing rules, and a set of export formats and destinations. Documents that are processed by workflows may come from monitored folders, email boxes, or SharePoint libraries.

Capability to Create Many Document Conversion Applications

The web service API makes it possible to process documents without having to save them to a shared folder, send them to an email box, or save them to a SharePoint library. This provides the capability to create an almost unlimited number of document conversion applications. This includes the conversion of documents within a document management system whose documents are kept under strict security and are accessible only through a secure API. An application could be created that would locate the desired documents in the document management system, submit them for processing to ABBYY Recognition Server, and check the processed documents back into the document management system. Other applications include the extraction of text from images that have been captured by smart phones using a mobile application. It would be possible to return the results to the user of the smart phone application while they wait. Another application might run within an Enterprise Resource Management(ERP) system and extract text and barcodes from a receipt, invoice, check images, or other business document as needed based on the ERP system determining that the document is not searchable.

Recognition Server Provides an Alternative to Cloud-Based OCR

ABBYY Recognition Server provides an alternative to cloud-based OCR solutions. The web service API allows documents to be processed in a fashion similar to cloud-based OCR solutions. This is especially attractive for sensitive applications such as medical or government where security and other regulations such as HIPAA would prevent the use of cloud-based OCR. ABBYY Recognition Server may be installed on servers in secure data centers or on customer-controlled secure cloud servers and security and regulation compliance maintained.

Recognition Server Features and Operations

ABBYY Recognition Server can perform the following operations through the web service API:

- Document conversion
  - Image (Image only PDF, JPEG, TIFF, BMP, etc.) to searchable formats such as PDF, PDF/A, Word, Text, or Excel
  - Office document (Word, Excel, PowerPoint, etc.) to searchable PDF or PDF/A
- Barcode extraction
  - Standard 1-d and 2-d barcodes including Code 39, 128, QR Code, and PDF417
  - Ability applies to all documents including Office type documents
- Text extraction
  - Ability to return text “on the fly”
  - Automatic language detection and the ability to separate the results by language
  - Provides text properties such as location by page and position within the page
  - Provides other text properties such as style, spacing, and font
  - Provides support for text extraction from tables with preservation of row and column information.
- Picture extraction
  - Ability to detect and pictures in documents
  - Ability applies to all documents including Office type documents
- Separator extraction
  - Ability to extract black and white or colored lines that are used to separate information within a document
  - Horizontal and vertical lines
  - Capture of position and size and any text that is contained within the separator
  - Could be utilized by advanced document analysis applications for locating information within a document

Unsupported Operations in Recognition Server

Every application has its limits, and although Recognition Server provides many advanced OCR features it does not provide support for either handwriting recognition (ICR) or optical mark recognition (OMR). Customers desiring these features would need to move to the more advance ABBYY FlexiCapture product or FlexiCapture Engine. Contact UFC for more information about these products.

Three other areas must be considered beyond the lack of ICR or OMR functions:

- ABBYY Recognition Server provides an "area template" feature that may be used to instruct the processing engine how to process the various text, image, and barcode zones found in the documents. But this must be constructed manually using either the ABBYY Recognition Server 4 Verification Station or ABBYY FineReader version 11 or earlier. This "area template" specifies the rules for processing or excluding the zones within the documents. But it is a hard-coded specification and templates must be created for each document shape. This is not meant to compete with products like ABBYY FlexiCapture that also provide templates but are meant to be used to specify data extraction rules for fields that are found within the documents. Area templates might be used to exclude certain parts of documents from processing or for changing the way that text is grouped into paragraphs for documents that follow a strict format.
- ABBYY Recognition server does not provide a REST API. The web service is based on SOAP.
- The ABBYY Recognition server web service does not return textual results directly through web service properties. What must be done is to process a document and request XML output. Then XML will contain the desired information.

About the Recognition Server Web Service API

The ABBYY Recognition Server web service API is a SOAP web service that runs under Microsoft ASP.NET. The virtual directory name is "Recognition4WS" for version 4 of the product. It can be expected that newer versions of the product will follow this same naming convention as previous versions have. Those virtual directories are still installed with the product and show up as "Recognition4WS.v1", "Recognition4WS.v2", and "Recognition4WS.v3." These correspond to versions of the web service for versions 1, 2, and 3 of the product and are meant to provide legacy compatibility. This document will refer to version 4 of the web service API.

The URL for the web service is: http://localhost/Recognition4WS/RSSoapService.asmx

Substitute the server name as appropriate with the proper server name or DNS name. Likewise, if a site certificate is being used, then swap "http" for "https." And if the default port has been changed from the standard port, enter that after the server or DNS name following a colon (server:port).

The default installation of ABBYY Recognition Server will establish Anonymous Authentication for the web service and require that users be authenticated by the ABBYY Recognition Server through the use of the Remote Administration Console using functions provided in the “Users” tab. This document will assume that user access has been established in advance using this utility.

Configuration of the Workflows

ABBYY Recognition Server uses workflows that are configured using the Remote Administration Console in order to specify the processing rules that apply for each set of documents to be processed. When using the web service API all that is necessary is for a single workflow to be configured with the bare minimum settings. These settings will be overridden by the application that is calling the web service. For extracting text from documents XML output will be requested and additional options will be specified for the XML output such that it includes the character formatting and in order to force a single XML output file for each page of the input document. For the rest of this document, a single workflow named “English” will be used. Before proceeding, setup at least one workflow and process some documents though it. An easy test would be to monitor a local folder on the server and pick up the files and create searchable PDF and XML output. Check the event log for any errors and verify that the output files look correct.

Getting Started with the Web Service API

The WSDL for the web service is available by adding “?wsdl” to the end of the URL in a browser. Either browse the WSDL URL from the desired development environment or save the WDSL locally to a file from a browser and then ingest that file. For this particular example, the WSDL was saved locally and a VB.NET wrapper class was created using the Visual Studio.NET wsdl.exe tool. This command may be launched from the VS developer command prompt. The first parameter should be the path to the wsdl file. Then use the /language option to select the language (cs for C Sharp, vb for VB, js for JavaScript, vjs for Visual J#, or cpp for C++). Finally use the /out parameter to specify the full path to the output class module. Use double quotes around directory names in case they have spaces. Example:

Wsdl.exe “C:\Development\RS Examples\RSSoapService.wsdl” /language:vb /out: “C:\Development\RS Examples\RSSoapService.vb”

Add this file into a new project. The top portion of the file that was added to the new Visual Studio project is shown below. Again, this was created in the VB.NET language so the new project should be a VB.NET project of some type such as a console application or like the example that will be shown here - a Windows forms application.

```
'''<remarks/>
<System.CodeDom.Compiler.GeneratedCodeAttribute("wsdl", "4.6.1055.0"),  _
 System.Diagnostics.DebuggerStepThroughAttribute(),  _
 System.ComponentModel.DesignerCategoryAttribute("code"),  _
 System.Web.Services.WebServiceBindingAttribute(Name:="RSSoapServiceSoap",
[Namespace]:="http://www.abbyy.com/RecognitionServer4_xml/RecognitionServer4.xml")>
_
Partial Public Class RSSoapService
```

Steps to Submit a Document to the Web Service API

For ease of example, add a button to the new Windows Forms application main form. The following sections will guide you through the steps that are necessary to submit a document for processing and to return the results. The steps shown could be implemented in a class or other module and do not present every option that is available with the web service API. The structure names such as “InputFile” refer to the definitions that were supplied by the web service through the WSDL which was consumed in the project. This example has been kept as simple as possible. So, a single file is processed even though it is possible to process many files in a single request. The number of changes to the various recognition setting has been kept to a minimum. A single recognition language is used in the example. Document conversion is showing using only the PDF/A format.

a. Instantiate the web service and connect to the server
b. Load a file from disk into an InputFile object
c. Create an XMLTicket and add specify the InputFile
d. Specify the recognition language
e. Specify the recognition options
f. For text and barcode extraction, specify XML output
g. For document conversion specify PDF/A output
h. Process the document
i. Access the results including the text and barcodes

The example assumes this import:

```
Imports System.IO
```

More details are shown for each step below.

Detailed Steps with Sample Code

a. Instantiate the web service and connect to the server

```
Dim objWebService As New RSSoapService
objWebService.Url =http://localhost/Recognition4WS/RSSoapService.asmx

Dim objCredentials As System.Net.ICredentials
objCredentials = New System.Net.NetworkCredential(“UsernameHere”,
“PasswordHere”)
objWebService.Credentials = objCredentials
```

b. Load a file from disk into an InputFile object

The path to the file should be provided to replace “path to file.” This is the image or other file that is desired to be processed and it must include the full path to the file.

```
Dim file__1 As New FileContainer
 Using stream As FileStream = System.IO.File.Open(“path to file”,
FileMode.Open)
                file__1.LocationPath = “path to file”
                file__1.FileContents = New Byte(stream.Length - 1) {}
                stream.Read(file__1.FileContents, 0,
file__1.FileContents.Length)
 End Using
```

c. Create an XMLTicket and add specify the InputFile

The “server name” is the name of the server. “English” is the name of the workflow that is being used. Not that all of the settings are going to be overridden in this workflow, but a workflow name is still required. It is possible to return an array of the workflow names using the web service call “GetWorkflows” and specifying the server name. The example could be modified to get the array of workflow names and then just use the first entry.

The name property on the XML ticket is set to the name of the input file because this property is used when naming the output files that are produced.

```
Dim objTicket As XmlTicket
Dim objInputFiles As InputFile()
objTicket = objWebService.CreateTicket(“server name”, “English”)
ReDim objInputFiles(0)
objInputFiles(0) = New InputFile
objInputFiles(0).FileData = file_1
objTicket.InputFiles = objInputFiles
objTicket.Name =
Path.GetFileName(objTicket.InputFiles(0).FileData.LocationPath)
```

d. Specify the recognition language

Multiple languages may be specified. The parameter is an array. Find the names of the languages using the Remote Administration Utility inside of the recognition settings for any workflow.

```
objTicket.RecognitionParams.Languages = {"English"}
```

e. Specify the recognition options

For this example barcodes will be read. No other options will be changed. The API does however support changing all of the settings that are available in the workflow configuration. This includes all of the options that are listed under the “Advanced Processing Settings” panel in the Process tab.

```
objTicket.RecognitionParams.LookForBarcodes = True
```

f. For text and barcode extraction, specify XML output

In order to retrieve the text and barcodes that were recognized, XML output is required. There are not distinct objects and properties for text and barcode values that are supported in the web service. One XML file will be requested per page of the source document. This is because the XML output does not include any reference to the page number within the source document. Restricting the output to a single page per XML file provides a means of tying the results back to the page number within the source document.

The example source code shown here shows some of the other properties that are available for the XML export. The extended character formatting option has been disabled here. That option provides additional information about the characters that have been recognized including confidence values and dictionary lookup values. See the ABBYY Recognition Server Developer's help file for each of these settings.

```
Dim objXMLExportSettings As New XMLExportSettings
objXMLExportSettings.WriteExtendedCharAttributes = False
objXMLExportSettings.WriteCharactersFormatting = True
objXMLExportSettings.WriteNonDeskewedCoordinates = False
objXMLExportSettings.PagesPerFile = 1
objXMLExportSettings.FileFormat = OutputFileFormatEnum.OFF_XML
```

g. For document conversion specify PDF/A output

This example shows the PDF/A export being setup such that the output includes the searchable text. And the PDF version is set to 1.7.

```
Dim objPDFExportSettings As New PDFAExportSettings
objPDFExportSettings.PDFExportMode = PDFExportModeEnum.PEM_ImageOnText
objPDFExportSettings.PdfVersion = PDFVersionEnum.PVN_Version17
```

h. Process the document

The first step is to connect the various export settings to the XML ticket. The export location must be provided. This will specify the directory for all of the exports. It is the responsibility of the user to clean up the processed files directory. The files will be named using unique names (GUID format) by the server. Note that the output filenames will not have file extensions.

Finally, the document is processed. This example was setup using synchronous processing, so the process calling the web service will wait until the processing has been completed. Then the IsFailed property on the XML result object may be checked for success or failure.

```
Dim formats As OutputFormatSettings()
ReDim formats(1)
formats(0) = objXMLExportSettings
formats(1) = objPDFExportSettings
objTicket.ExportParams.ResultLocationPath = "C:\ABBYYRS\Output\XML"
objTicket.ExportParams.Formats = formats
Dim objXMLResult As XmlResult
objXMLResult = objWebService.ProcessTicket("server name", "English")
If objXMLResult.IsFailed Then
  ' Do something here… the processing failed….
End If
```

i. Access the results including the text and barcodes

The XML Result object provides access to the filenames that were produced for each export type. Since a single document was submitted, a single “JobDocuments” result is returned and this element is referenced using the index of 0. If multiple documents were submitted then the JobDocuments array would contain one element for each input document.

An elegant way that UFC uses to process the XML results that come back for each document is to create a wrapper for the FineReader 10 XML schema that is used for the XML results. That is not being shown here. There is a tiny amount of documentation for the schema provided in the ABBYY Recognition Server Developer’s help file under the section “Document XML Scheme.” The typo on the word “schema” may have been intentional since very little information is provided. The XML provides information about each block that is recognized with block types being one of the following types: Text, table, picture, barcode, or separator.

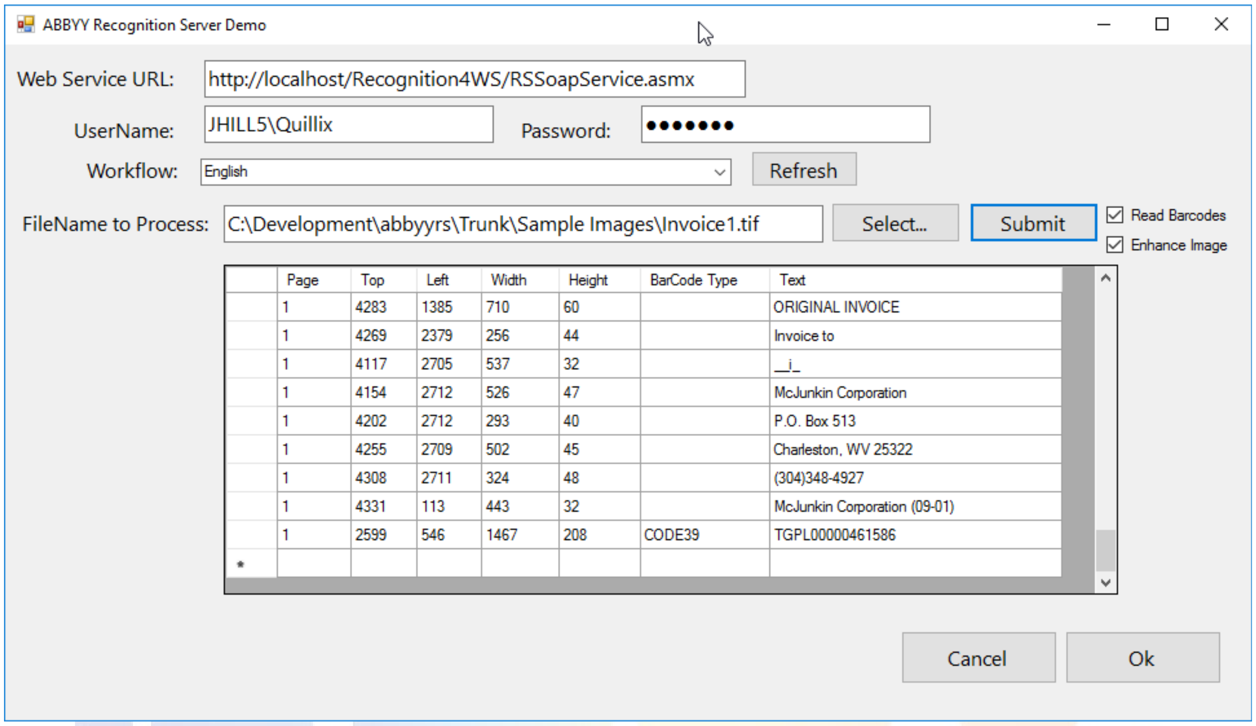
```
Dim iPageNumber As Integer
If objXMLResult.JobDocuments(0).OutputDocuments.Length > 0 Then]
 iPageNumber = 1
 For Each oOutputDocumentFile As OutputDocument In
objXMLResult.JobDocuments(0).OutputDocuments
    If oOutputDocumentFile.FormatSettings.FileFormat = OutputFileFormatEnum.OFF_XML Then

      ‘
      ‘ Process each XML file
      ‘ The filename for the XML file is: objFileContainer.LocationPat
      ‘Once the text and barcode values are read from this file it should be deleted
      System.IO.File.Delete(objFileContainer.LocationPath)
    End If
    If oOutputDocumentFile.FormatSettings.FileFormat = OutputFileFormatEnum.OFF_PDFA
Then
     'Do something with the PDF/A file whose path is objFileContainer.LocationPath
     'If the file was not moved, make sure to clean it up
     System.IO.File.Delete(objFileContainer.LocationPath)
    End If
 iPageNumber += 1

 Next
End If
```

Example Result

The picture below shows the screen from a slightly more complex application that was built using the ABBYY Recognition Server web service. The steps used were matched the steps outlined here exactly except the XML results were populated to a grid and the block type was respected such that barcodes were displayed differently than text blocks. There was a single barcode found in the test document that was used. The PDF/A document that was produced by the program would have been saved to a file server.



Web Service URL: http://localhost/Recognition4WS/RSSoapService.asmx

UserName: JHILL5\Quillix Password: ●●●●●●●

Workflow: English

FileName to Process: C:\Development\abbyyrs\Trunk\Sample Images\Invoice1.tif

☑ Read Barcodes
☑ Enhance Image

Page	Top	Left	Width	Height	BarCode Type	Text
1	4283	1385	710	60		ORIGINAL INVOICE
1	4269	2379	256	44		Invoice to
1	4117	2705	537	32		__i_
1	4154	2712	526	47		McJunkin Corporation
1	4202	2712	293	40		P.O. Box 513
1	4255	2709	502	45		Charleston, WV 25322
1	4308	2711	324	48		(304)348-4927
1	4331	113	443	32		McJunkin Corporation (09-01)
1	2599	546	1467	208	CODE39	TGPL00000461586

Conclusion

The ABBYY Recognition Server web service API may be used to convert documents into searchable, archival formats. It may also be used to extract text and barcodes from documents. It provides a more secure option for sensitive documents over cloud-based OCR solutions and may be integrated into any application that is capable of calling a web service. ABBYY Recognition Server provides access to the text and barcodes that were found in each document that was submitted for processing through the web service. And the results include all of the location information for each block of text or barcode.

How User Friendly Consulting Can Help

Contact us regarding purchasing ABBYY Recognition Server, watching a live demo with your sample documents, or even obtaining a copy of the software for testing on your own server. The example shown uses the web service API which is one of only a couple of add-on features. Also let us know if you intend to process Chinese, Japanese, or Korean documents.

We provide consulting services for ABBYY Recognition Server, including both the installation and administration of the product well as assisting with or performing development using the web service

API. Please reach out to us if you would like to have us solve your document conversion or data extraction challenge or just for help with properly configuring your existing ABBYY Recognition Server system. We provide a wide range of consulting options including ABBYY trained and certified personnel as well as a wide range of training options to get your employees up to speed on the product very quickly. We also distribute other ABBYY products such as ABBYY FlexiCapture that are more appropriate for field-based document data extraction.