XImage.OCR SDK Developer Guide

Table of Content

Set OCR Resource Path

Set path of the training resources

C#

```
//  the folder that contains all language '.traineddata' files
String folderPath = @"...";
OCRHandler.SetTrainResourcePath(folderPath);
```

Scan an Image File (JPEG)

Using OCRHandler to process the image file

C#

```
// set training resource path
OCRHandler.SetTrainResourcePath(OCR_SOURCE_FOLDER_PATH);

String inputFilePath = Program.ROOT_PATH + "\\" + @"1.jpg";
String outputFilePath = Program.ROOT_PATH + "\\" + @"output.txt";

// scan the input file and output the result to a TXT file
int errCode = OCRHandler.Translate(inputFilePath, MIMEType.TXT, outputFilePath);
if (errCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Using OCRPage to import the image

C#

```
// set training resource path
OCRHandler.SetTrainResourcePath(OCR_SOURCE_FOLDER_PATH);

String inputFilePath = Program.ROOT_PATH + "\\" + @"1.jpg";

// create a Bitmap object
Bitmap image = new Bitmap(inputFilePath);

// create an OCRPage
OCRPage page = OCRHandler.Import(image);
// do recognize and get the error code
if (page.Recognize() == 0)
{
   // get scan result in String
   Console.WriteLine(page.GetText());
}
```

Scan an TIFF File

Using OCRHandler to process the file

C#

```
// set training resource path
OCRHandler.SetTrainResourcePath(OCR_SOURCE_FOLDER_PATH);

String inputFilePath = Program.ROOT_PATH + "\\" + @"1.tif";
String outputFilePath = Program.ROOT_PATH + "\\" + @"output.txt";

// scan the input file and output the result to a TXT file
int errCode = OCRHandler.Translate(inputFilePath, MIMEType.TXT, outputFilePath);
if (errCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Remark

Must to add reference “RasterEdge.XDoc.TIFF.dll” for this feature.

Using OCRPage to import the file

C#

```
// set training resource path
OCRHandler.SetTrainResourcePath(OCR_SOURCE_FOLDER_PATH);

String inputFilePath = Program.ROOT_PATH + "\\" + @"1.tif";
// load .tif file
TIFFDocument doc = new TIFFDocument(inputFilePath);
// scan all pages in the file
for (int pageIndex = 0; pageIndex < doc.GetPageCount(); pageIndex++)
{
   BasePage tifPage = doc.GetPage(pageIndex);
   // import TIFF page
   OCRPage page = OCRHandler.Import(tifPage);
   // do recognize and get the error code
   if (page.Recognize() == 0)
   {
      Console.WriteLine("Page " + pageIndex + ": ");
      // get scan result in String
      Console.WriteLine(page.GetText());
   }
}
```

Remark

Must to add reference “RasterEdge.XDoc.TIFF.dll” for this feature.

Scan a PDF File

Using OCRPage to import the file

C#

```
// set training resource path
OCRHandler.SetTrainResourcePath(OCR_SOURCE_FOLDER_PATH);

String inputFilePath = Program.ROOT_PATH + "\\" + @"1.pdf";
// load .pdf file
PDFDocument doc = new PDFDocument(inputFilePath);
// scan all pages in the file
for (int pageIndex = 0; pageIndex < doc.GetPageCount(); pageIndex++)
{
   BasePage pdfPage = doc.GetPage(pageIndex);
   // import TIFF page
   OCRPage page = OCRHandler.Import(pdfPage);
   // do recognize and get the error code
   if (page.Recognize() == 0)
   {
      Console.WriteLine("Page " + pageIndex + ": ");
      // get scan result in String
      Console.WriteLine(page.GetText());
   }
}
```

Remark

Must to add reference “RasterEdge.XDoc.PDF.dll” for this feature.

Output Scan Result

Output scan result as PDF file

C#

```
// set training resource path
OCRHandler.SetTrainResourcePath(OCR_SOURCE_FOLDER_PATH);

String inputFilePath = Program.ROOT_PATH + "\\" + @"1.tif";
String outputFilePath = Program.ROOT_PATH + "\\" + @"output.pdf";

// scan the input file and output the result to a PDF file
int errCode = OCRHandler.Translate(inputFilePath, MIMEType.PDF, outputFilePath);
if (errCode == 0)
{
   Console.WriteLine("Success");
}
else
{
   Console.WriteLine("Failed");
}
```

Remark

Must to add reference “RasterEdge.XDoc.PDF.dll” for this feature.

FAQ:

1. Application running on non-English operating systems.

Please add the following code to the beginning of your program:

```
System.Threading.Thread.CurrentThread.CurrentCulture = new System.Globalization.CultureInfo("en-US", true);
```