Automatically Classify Scanned Documents using LEADTOOLS Forms Recognition

LEADTOOLS®
THE WORLD LEADER IN IMAGING SDKs

LEAD TECHNOLOGIES
INCORPORATED

1

Introduction

Document imaging is certainly saving trees and physical storage space, but in some situations it fails to save much time or hassle. Opting in to paperless statements or manually scanning paper documents yourself is a great way to archive all of your bills, invoices, financial statements and the like. However, it still requires a fair amount of time and energy to thoughtfully organize the documents in your digital filing cabinet. After all, what good is it to digitally archive your documents if they become near impossible to find them again in the future?

That scenario might not be too overwhelming for an individual with a good memory and habits, but what about medium to large businesses and corporations that deal with thousands of documents on a daily basis and has hundreds of people working with the same digital archive? Without some kind of automation, you have a huge overhead of man-hours and – what's even more problematic – a wide opportunity for human error.

Imagine being able to drop all of your scanned documents into a single folder and have all the work of moving and renaming the files in a logical, consistent manner done automatically. LEADTOOLS Forms Recognition and Processing fits the bill perfectly with its high level, flexible and powerful imaging libraries. Applications built with LEADTOOLS can compare a scanned document against known templates and correctly classify the document type. After a document is correctly identified, LEADTOOLS can then extract OCR, OMR, Barcodes and more from defined locations on the form.

Processing the Document Repository

The first step to solving this dilemma is to process and manage a central location where all of the scanned documents are placed for classification. There are multiple ways of accomplishing this, such as using a web service or windows service or a monkey pressing a button. The method chosen in this example is a simple console application which is then scheduled to run with Windows' task scheduler.

The code that manages the repository is relatively simple since it primarily uses basic file and folder operations with the `System.IO` namespace. However the most crucial part of the application is passed on to the `DocumentClassifier` which encapsulates the LEADTOOLS Forms Recognition features to return the data used for moving and renaming the documents.

You can download the entire project from the LEADTOOLS support forums at this link:
https://www.leadtools.com/support/forum/posts/t12307-1

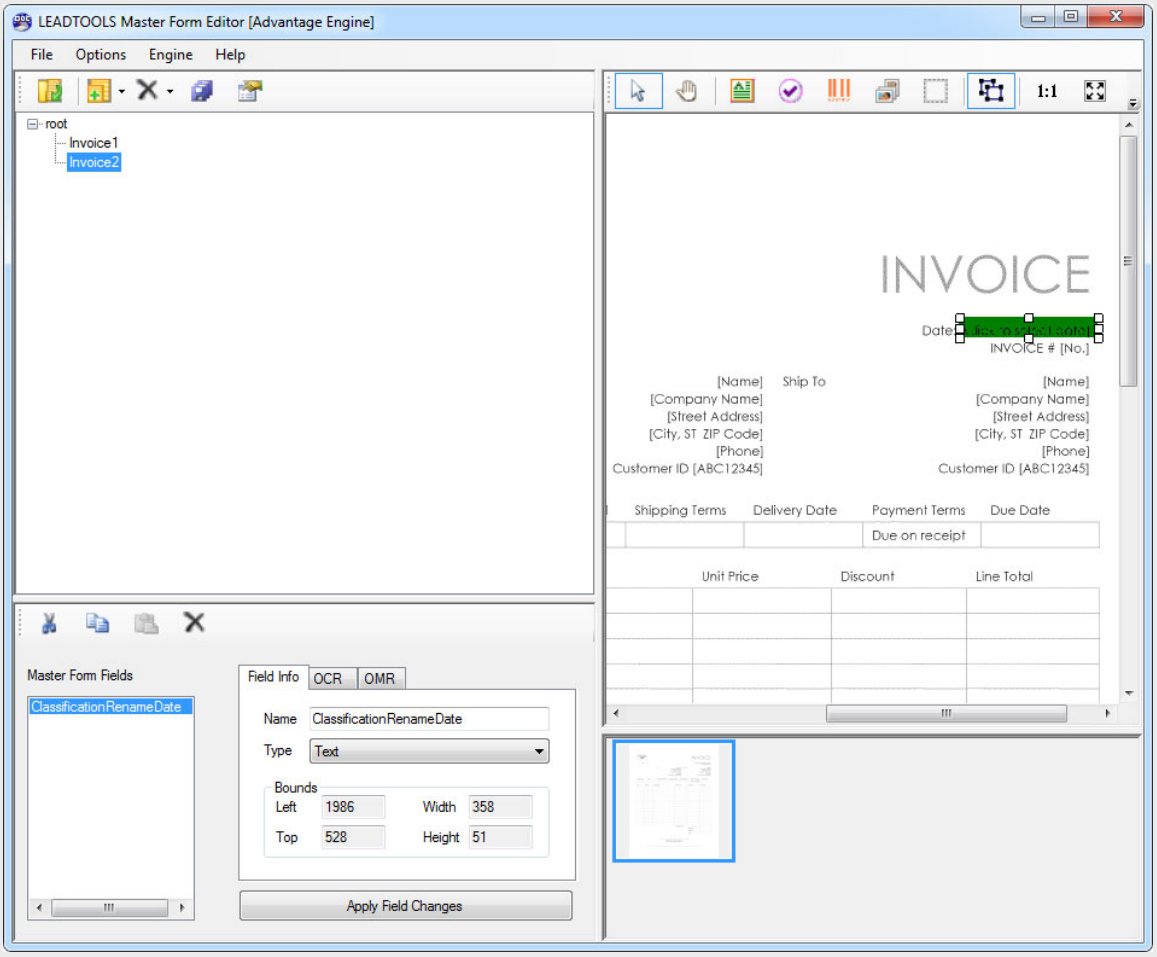
```
// Check the scanned document repository for new documents
string[] newDocuments = Directory.GetFiles(docRepositoryNewDocs);
DocumentClassifier docClassifier = new
   DocumentClassifier(docRepositoryMasterForms);
string movedDocumentName, masterFormSubFolder;
foreach (string currentDoc in newDocuments)
{
   movedDocumentName = null;
   // Try to match this document against known document types
   ClassifiedDocument classifiedDoc =
      docClassifier.ClassifyDocument(currentDoc);
   if (classifiedDoc.MasterFormName != null)
   {
      // Add the subfolder for the master form if it doesn't exist
      masterFormSubFolder = string.Format("{0}{1}\\", docRepositoryRoot,
         classifiedDoc.MasterFormName);
      if (!Directory.Exists(masterFormSubFolder))
         Directory.CreateDirectory(masterFormSubFolder);

      // rename the file according to the date found
      if (classifiedDoc.DocumentDate != DateTime.MinValue)
      {
         movedDocumentName = string.Format("{0}{1}{2}",
            masterFormSubFolder,
            classifiedDoc.DocumentDate.ToString("yyyyMMdd"),
            currentDoc.Substring(
               currentDoc.LastIndexOf('.'),
               currentDoc.Length - currentDoc.LastIndexOf('.')));
      }
      else
      {
         // Didn't find a date to rename with, so just move it
         movedDocumentName = currentDoc.Replace(docRepositoryNewDocs,
            masterFormSubFolder);
      }
   }
   else
   {
      movedDocumentName = currentDoc.Replace(docRepositoryNewDocs,
         docRepositoryUnclassifiedDocs);
   }

   if (!string.IsNullOrEmpty(movedDocumentName))
      File.Move(currentDoc, movedDocumentName);
```

3

Using LEADTOOLS Forms Recognition

Before LEADTOOLS can start classifying documents it must know how to classify them, which is accomplished by creating a collection of Master Form templates. LEADTOOLS ships with a Master Form editor demo which we will use to add a master form for two different invoices containing a single OCR field that extracts the invoice date which can be used to rename the file.

Figure 1: Defining Master Form Templates with the Master Form Editor



4

Now that our master forms are defined, we are ready to process the documents. We have scanned two filled out invoices based on the master forms, and a tax form which does not have a known template. For each file in the "New" folder, LEADTOOLS will compare it against the master templates. If a match is found, it will then process the document's fields and return the form's name and the date field.

```
// Create an OCR Engine for each processor on the machine. This
// allows for optimal use of thread during recognition and processing.
ocrEngines = new List<IOcrEngine>();
for (int i = 0; i < Environment.ProcessorCount; i++)
{
   ocrEngines.Add(OcrEngineManager.CreateEngine(
      OcrEngineType.Advantage, false));
   ocrEngines[i].Startup(formsCodec, null, String.Empty, String.Empty);
}
// Point repository to directory with existing master forms
formsRepository = new DiskMasterFormsRepository(formsCodec,
   _MasterFormFolder);
autoEngine = new AutoFormsEngine(formsRepository, ocrEngines, null,
   AutoFormsRecognitionManager.Default | AutoFormsRecognitionManager.Ocr,
   30, 70, true);

// Run the forms recognition on this document
AutoFormsRunResult runResult = autoEngine.Run(document, null);
if (runResult != null)
{
   // In this example we use two pieces of information
   // to organize the classified forms:
   // 1. Form name is used for the sub folder
   // 2. "ClassificationRenameDate" field for the file name
   retClassifiedDocument.MasterFormName =
      runResult.RecognitionResult.MasterForm.Name;

   // Process the recognized form and extract desired info
   foreach (FormPage formPage in runResult.FormFields)
   {
      foreach (FormField field in formPage)
      {
         if (field != null && field.Name == "ClassificationRenameDate")
         {
            retClassifiedDocument.DocumentDate =
               DateTime.Parse((field.Result as TextFormFieldResult).Text);
         }
      }
   }
}
```

Figure 2: Forms Repository and Sub Folders Before Classification

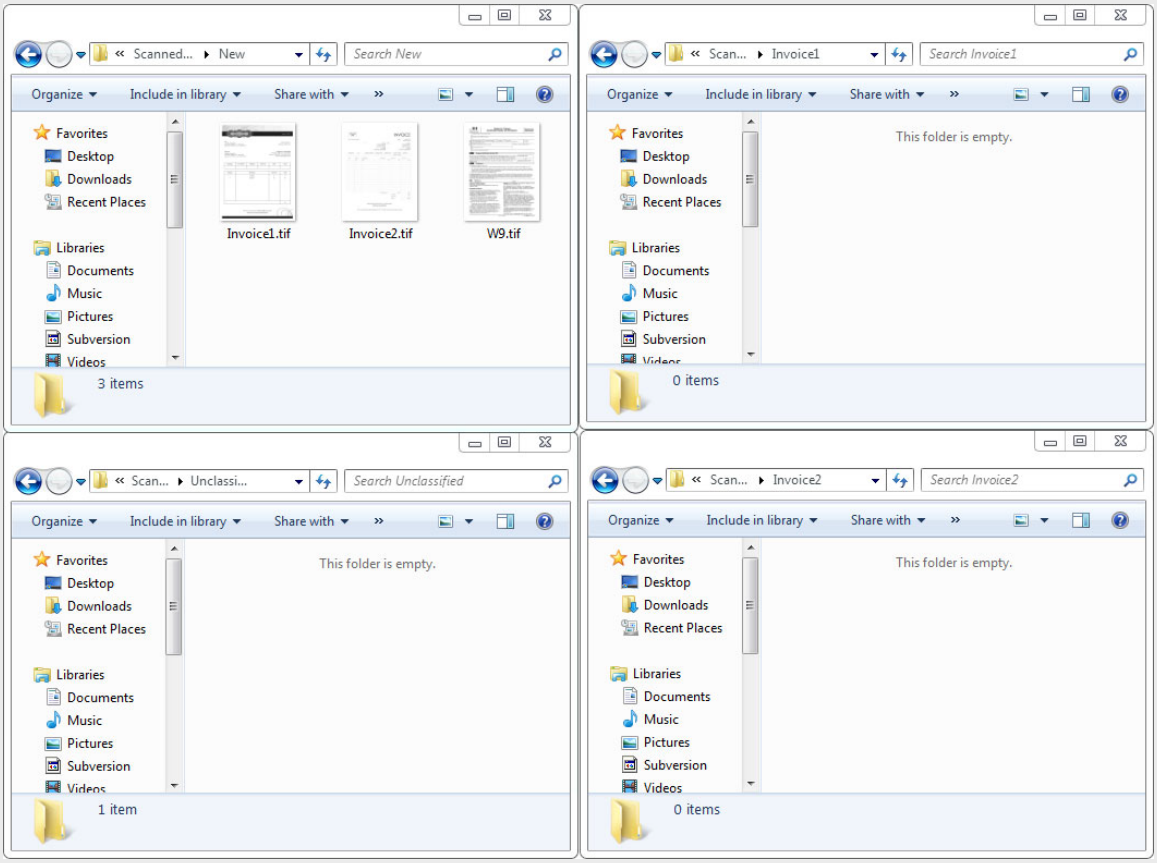
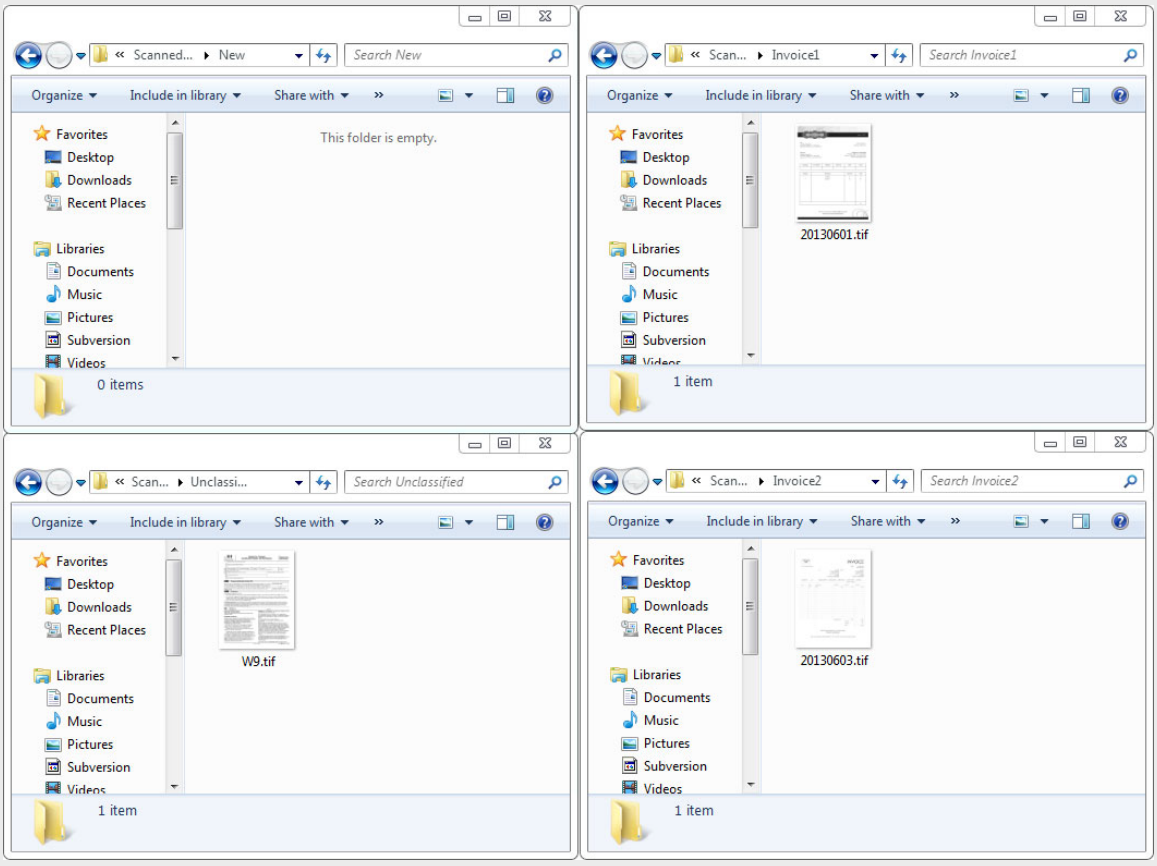


Figure 3: Forms Repository and Sub Folders After Classification



As you can see, the two invoices were correctly matched to their master form and renamed based on the date field. Additionally, the unclassified documents folder acts as a fail-safe, letting the application grow and adapt with minimal effort. When you have a new document type that is not in your master forms collection, all you have to do is use one of those images as a template, add the fields you want to extract, and move the unclassified documents back into the New folder to get processed again the next time the application runs.

Taking it a Step Further

This simple solution has massive potential for expansion and adaptation. For example, you could easily manage your documents online by connecting to a cloud service such as Google Docs, SkyDrive or iCloud. Similarly, businesses could adapt it to monitor and organize incoming faxes and email attachments or use the recognized field data and store it directly into a database. Most importantly, LEADTOOLS Forms Recognition can process as much or as little information from the scanned documents as you desire, stretching its usefulness far beyond mere organization and archival. Form fields, check boxes, invoice amounts, and much more can be extracted to speed up any workflow.

Conclusion

This is just one of many real world solutions you can tackle with LEADTOOLS. Its state of the art Forms Recognition and Processing SDK the most flexible and powerful product in its class, and LEADTOOLS offers an incredible value with its comprehensive family of toolkits for raster, document, medical and multimedia imaging. For more information on how LEAD Technologies can image-enable your application and boost your ROI, visit ***www.leadtools.com*** to download a free evaluation, or give us a call at +1-704-332-5532.

SALES: (704) 332-5532
SALES@LEADTOOLS.COM

SUPPORT: (704) 372-9681
SUPPORT@LEADTOOLS.COM

LEAD TECHNOLOGIES, INC.
1927 SOUTH TRYON STREET
SUITE 200
CHARLOTTE, NC 28203

About LEAD Technologies

With a rich history of over twenty years, LEAD has established itself as the world's leading provider of software development toolkits for document, medical, multimedia, raster and vector imaging. LEAD's flagship product, LEADTOOLS, holds the top position in every major country throughout the world and boasts a healthy, diverse customer base and strong list of corporate partners including some of the largest and most influential organizations from around the globe.

LEADTOOLS®
THE WORLD LEADER IN IMAGING SDKs

LEAD TECHNOLOGIES
INCORPORATED