

# Creating and Processing OMR Forms with **LEADTOOLS**

---

## Introduction

---

Forms recognition and processing is used all over the world to tackle a wide variety of tasks including classification, document archival, optical character recognition and optical mark recognition. Out of those general categories, OMR is an oft misunderstood and underused feature in document imaging due to the time required to set up OMR based forms and the difficulty of accurately detecting which OMR fields are filled on a scanned document. Creating and processing OMR forms can be a time-consuming nightmare and this white paper will discuss how to alleviate those issues through automated detection, classification and processing.

Most forms contain a small number of OMR fields to capture information such as gender and marital status. These cause little to no difficulties because there are very few fields to deal with. On the other hand, creating and processing forms dominated by multiple choice questions is noticeably more difficult due to the sheer volume of fields that can be packed into a page. Additionally, the small size of check boxes, bubbles and other types of OMR fields creates potential hypersensitivity resulting in more false negatives or positives.

Below we will examine in more detail how to alleviate both of these common problems by developing an OMR forms recognition application with LEADTOOLS. This award-winning imaging SDK contains all the tools necessary to combine time-saving and programmer friendly APIs with state of the art recognition accuracy and speed for an unmatched level of quality in your final solution.

---

## Using LEADTOOLS OCR to Add OMR Fields to a Master Form

---

The first step in a forms recognition application is to build the master forms. These master forms, or blank form templates, serve two primary purposes. First, it is used to identify what type of form a scanned document is. Second, the fields indicate the areas on the form from which data will be recognized and extracted.

For many systems, creating an OMR based form can be a tedious process due to the amount of repetition involved with surveys, bubble sheets or tests. One could spend hours manually drawing each and every OMR field around the boxes. Thankfully, LEADTOOLS is capable of automatically detecting all of the OMR fields with its [IOcrEngine.AutoZone](#) function. After finding each zone on the page, you can loop through the collection and add a new OMR field for each OMR zone.

```
FormPages formPages = currentMasterForm.ReadFields();

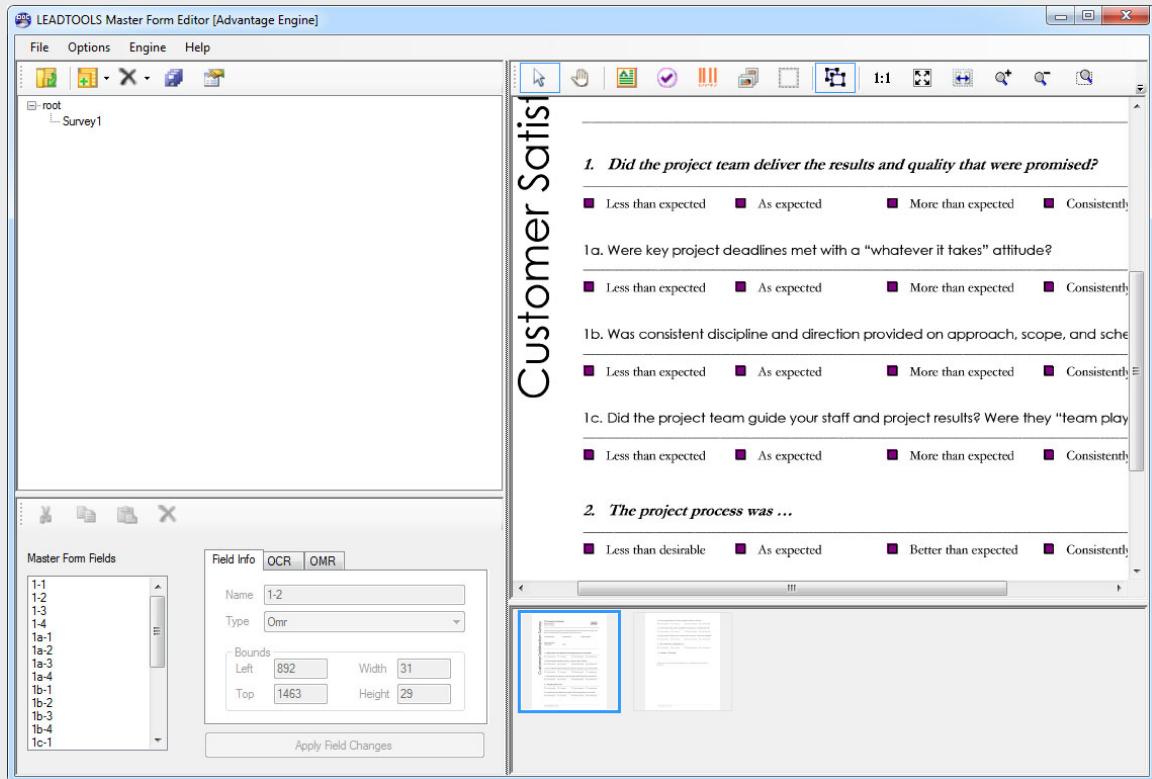
// Create OCR Engine
using (IOcrEngine ocrEngine =
    OcrEngineManager.CreateEngine(OcrEngineType.Advantage, false))
{
    ocrEngine.Startup(null, null, null, null);
    ocrEngine.SettingManager.SetEnumValue("Recognition.Zoning.Options",
        "Detect Text, Detect Graphics, Use Text Extractor, " + _
        "Detect Checkbox");

    using (IOcrDocument ocrDocument =
        ocrEngine.DocumentManager.CreateDocument())
    {
        // Auto zone
        ocrDocument.Pages.AddPages(rasterImageViewer1.Image, 1, 1, null);
        ocrDocument.Pages.AutoZone(OcrZoneParser.Leadtools,
            OcrZoneFillMethod.Omr, LogicalUnit.Pixel, 0, 0, null);

        // Add a form field for each OMR zone

        FormField newField;
        IOcrZoneCollection zones = ocrDocument.Pages[0].Zones;
        for (int i = 0; i < zones.Count; i++)
        {
            if (zones[i].FillMethod == OcrZoneFillMethod.Omr)
            {
                newField = new OmrFormField();
                newField.Bounds = zones[i].Bounds;
                newField.Name = string.Format("OMR Field {0}", i);
                formPages[oldSelectedPageIndex].Add(newField);
            }
        }
    }
}

currentMasterForm.WriteFields(formPages);
}
```



**Figure 1: Master Forms Editor after OMR Field Detection**

The OCR engine's [AutoZone](#) method is used to get the location of each zone but there are many ways to go about naming them. This simple example gives a base name to the zones, but one could expand on this logic and name the zones more intelligently by checking the [FormField.Bounds](#) property to determine which zones are in the same row or column. Additionally, you can use the Master Forms Editor demo or manually edit the XML file in which the field data is stored.

## Using LEADTOOLS Forms Recognition and Processing

Most scanned document processing systems must handle more than one type of form. A viable but inefficient solution might utilize a different application, button or dialog for each type of form that needs processing. This could certainly be implemented to automate the processing of data, but is handicapped by the requirement of manually informing the application which form template to process the scanned image with. An optimal solution is one in which the forms can be recognized or classified automatically and then processed based on those findings. LEADTOOLS provides reliable and flexible Forms Recognition capabilities with a variety of classification data including logos, dark and light areas, OCR, barcode and more.

```

// 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, masterFormsFolder);
autoEngine = new AutoFormsEngine(formsRepository, ocrEngines, null,
    AutoFormsRecognitionManager.Default | AutoFormsRecognitionManager.Ocr,
    30, 80, true);

string[] formsToRecognize = Directory.GetFiles(filledFormsFolder);
progressBar1.Maximum = formsToRecognize.Length;
for (int i = 0; i < formsToRecognize.Length; i++)
{
    // Recognize (Classify) the form
    lblStatus.Text = string.Format("Recognizing form {0} of {1}",
        i + 1, formsToRecognize.Length);
    AutoFormsRunResult runResult = autoEngine.Run(formsToRecognize[i], null);
    if (runResult != null)
    {
        // Recognition was successful
        lblStatus.Text = string.Format("Processing form {0} of {1}",
            i + 1, formsToRecognize.Length);
        ProcessResults(runResult);
    }

    progressBar1.Value++;
}

```

---

## Extracting Answers from Completed OMR Forms

---

Once the form is recognized successfully, the fields can be processed to extract the OMR data from the filled out document. An important consideration when choosing an OMR solution is how accurately it can handle variances in fill styles. Even if strict rules are communicated to those filling out the forms, there will still be differences in how humans fill in the OMR fields. LEADTOOLS excels in its OMR accuracy and can distinguish between filled and unfilled boxes regardless of fill styles. For example, see the following screen captures of the same question from three filled surveys.

1. *Did the project team deliver the results and quality that were promised?*

Less than expected     As expected     More than expected     Consistently more

1. *Did the project team deliver the results and quality that were promised?*

Less than expected     As expected     More than expected     Consistently more

1. *Did the project team deliver the results and quality that were promised?*

Less than expected     As expected     More than expected     Consistently more

**Figure 2: Differences in How OMR Fields Are Filled**

If you recall Figure 1, you can see the fields were named with the question number and column number separated by a hyphen. Armed with that naming paradigm we can then easily determine which checkbox was filled for each column and add it to our data source.

```
int nNewRowIndex = dataGridView1.Rows.Add();
foreach (FormPage formPage in runResult.FormFields)
{
    foreach (FormField field in formPage)
    {
        if (field.Result.GetType() == typeof(OmrFormFieldResult))
        {
            // Was this checkbox filled?
            if ((field.Result as OmrFormFieldResult).Text == "1")
            {
                // Get the question number and value (column number)
                // of this checkbox
                string[] strQuestionValue = field.Name.Split('-');
                dataGridView1.Rows[nNewRowIndex].Cells[string.Format("col{0}",
                    strQuestionValue[0])].Value = strQuestionValue[1];
            }
        }
    }
}
```

	1	1a	1b	1c	2	2a	2b	2c	2d	3
▶	3	4	3	2	3	2	3	4	4	3
	4	4	3	4	3	3	3	4	4	4
	1	2	2	1	2	2	1	1	2	1

Done!

Process Filled Forms

*Figure 3: Results from Completed Surveys*

Naturally, there are many ways to name the fields and correlate the answers to your data source. With a little planning at the beginning stages of your application, you can design your OMR Forms recognition solution around any master form and data source for a dependable, flexible and most importantly, accurate solution using LEADTOOLS.

---

## 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 is 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](http://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**<sup>®</sup>  
THE WORLD LEADER IN IMAGING SDKs

 **LEAD**  
**TECHNOLOGIES**  
I N C O R P O R A T E D