HTML5 Web Scanning with LEADTOOLS

LEADTOOLS®
THE WORLD LEADER IN IMAGING SDKs

LEAD TECHNOLOGIES INCORPORATED

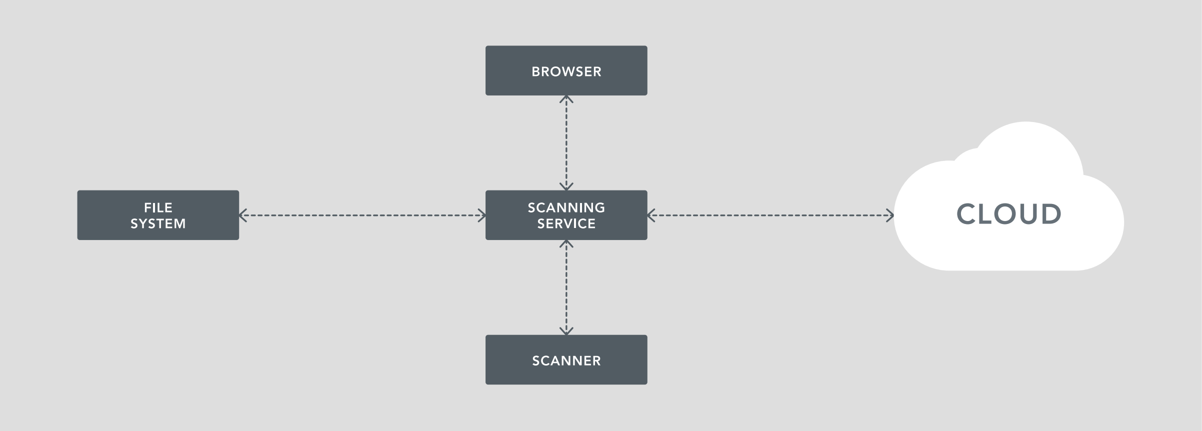
Introduction

Scanning and document capture is an essential aspect and starting point of most document imaging applications. For web developers, this is a problematic requirement due to the localized, hardware-dependent nature of scanners and image capture devices. Many roadblocks await the developer and most often the client is left with tedious and time-wasting inconveniences that make the application difficult to use. For example, scanning in a separate application creates a laborious workflow for the client. Browser plug-ins limit the flexibility of your application by locking it into specific browsers. Last but not least, updating and maintaining multiple codebases can turn your web-based scanning application into a nightmare.

LEADTOOLS has developed a Web Scanning SDK that solves many of these obstacles and offers a programmer-friendly and customizable framework for integrating both TWAIN (Windows) and Sane (Linux) scanning into a single web-based document imaging solution. In this white paper, we will explore the inner-workings of the Windows version of this innovative framework and highlight how developers can write just a few lines of code to extend and customize it for their application.

LEADTOOLS Web Scanning Components and Architecture

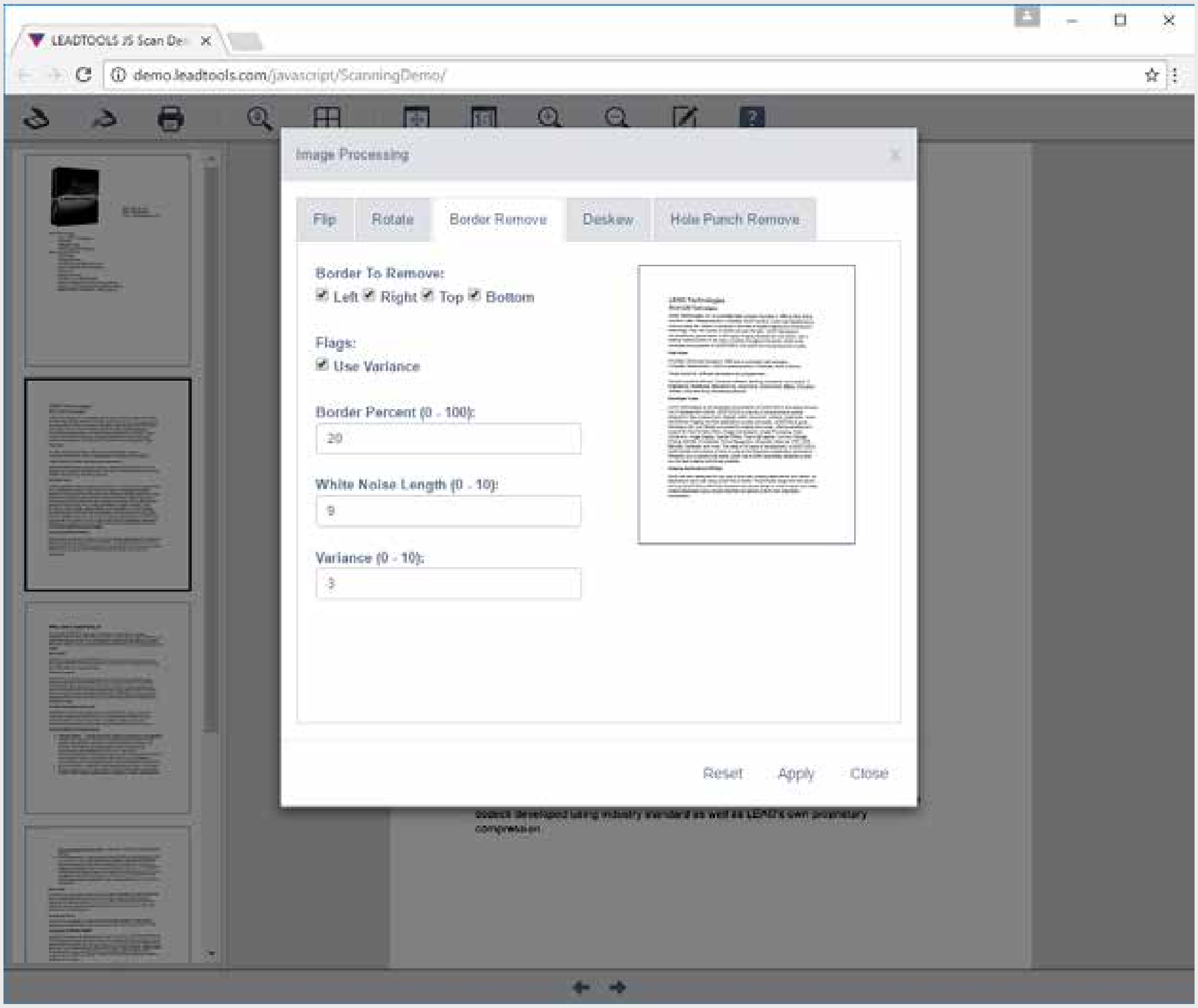
LEADTOOLS Web Scanning consists of two major components: a cross-browser web interface built with HTML5 and JavaScript, and a Self-hosted Web Service. Each of these components is designed with maximum extensibility in mind and makes it possible to create a robust web application with simple update procedures and separation of concerns for multi-developer teams.



The flexibility of this architecture allows developers to utilize one or more components for new projects, or for adding web scanning functionality to their existing web enterprise solution.

HTML5/JavaScript Web Application

The client drives the application with a robust web interface that uses LEADTOOLS HTML5 Image Viewing controls and communicates with the SHWS through RESTful JavaScript commands. The demo that ships with LEADTOOLS includes the thumbnail browser, image viewer, and several client-side tools including pan, zoom, fit, stretch, and magnifying glass. Additional features can easily be added including annotations and markup, drag and drop, and more. Best of all, since this application is designed using HTML5 and JavaScript, it can run without any browser plug-ins on any supported browser on a Windows or Linux machine running the SHWS.



Border To Remove:
☑ Left ☑ Right ☑ Top ☑ Bottom

Flags:
☑ Use Variance

Border Percent (0 - 100): 20

White Noise Length (0 - 10): 9

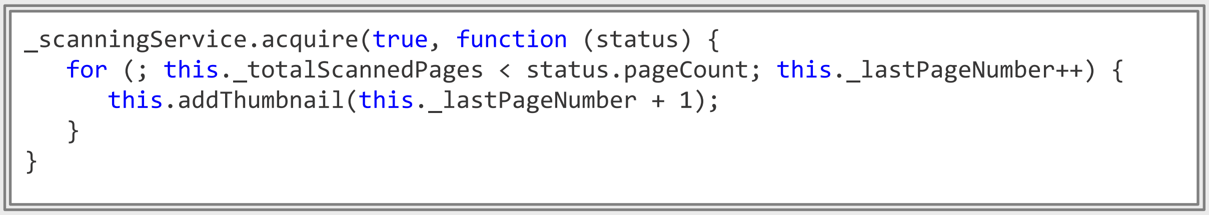
Variance (0 - 10): 3

Figure 1: Performing image processing on scanned images

3

Self-hosted Web Service

At the heart of the LEADTOOLS Web Scanning application is the ScanningService, a Self-hosted Web Service (SHWS). This is a necessary component, since the TWAIN device's drivers can only be installed on the client machine. The SHWS receives REST commands via JavaScript that allow the browser client to select the TWAIN source, negotiate TWAIN capabilities, get the status, and acquire images to a local cache which the web application can load via URI.



```
_scanningService.acquire(true, function (status) {
   for (; this._totalScannedPages < status.pageCount; this._lastPageNumber++) {
      this.addThumbnail(this._lastPageNumber + 1);
   }
}
```

Second to negotiating with the scanner, another primary function of the SHWS is applying image processing to the image. Its extensible interface builds upon a CommandCallback which can be used to implement any of LEADTOOLS' 200+ image processing functions. On the client side, calling the ApplyImageProcessingCommand JavaScript function will trigger the SHWS Run function which will apply the image processing function and parameters to the image and return the image to the client.

```
public static void Run(PageImageProcessingEventArgs e)
{
   string commandName = e.CommandName;
   string arguments = e.Arguments;
   RasterImage image = e.Image;

   if (e.IsPreview)
      DemoUtils.ResizeImage(image, e.PreviewWidth, e.PreviewHeight);

   // check the commands
   var ipFunction = _ipCommands[commandName] != null ?
      _ipCommands[commandName] : null;
   if (ipFunction != null)
      ipFunction(image, arguments);
}
// Dictionary mapping list of IP commands to their respective functions
private static readonly IDictionary<string, IPFunction> _ipCommands =
   new Dictionary<string, IPFunction>
{
   { “Flip”, FlipImage },
   { “Rotate”, RotateImage },
   { “Deskew”, DeskewImage },
   { “HolePunchRemove”, HolePunchRemove },
   { “BorderRemove”, BorderRemove }
};
//...
private static void DeskewImage(RasterImage image, string arguments)
{
   int angleRange = DemoUtils.JsonStringToInteger(arguments, “angleRange”);
   int angleResolution = DemoUtils.JsonStringToInteger(
      arguments, “angleResolution”);
   RasterColor fillColor = DemoUtils.JsonStringToRasterColor(arguments, “fillColor”);
   int flags = DemoUtils.JsonStringToInteger(arguments, “flags”);

   DeskewCommand cmd = new DeskewCommand(fillColor, (DeskewCommandFlags)flags);
   cmd.AngleRange = angleRange;
   cmd.AngleResolution = angleResolution;

   cmd.Run(image);
}
```

5

Further Customizations with the Command Callback

The CommandCallback can be used for more than just image processing. In fact, the callback is designed with full control and customization in mind. Anything you want to code into the SHWS is a possibility: save images to local folder, upload images to cloud storage, OCR, Barcode, etc. The simple example below shows how to save the scanned image to the local drive.

JavaScript Code

```
private saveImageBtn_Click(e: JQueryEventObject): void {
   var param: any = new Object();
   param.filename = GetFileName(); // Get from dialog
   param.format = GetFileFormat(); // Get from dialog
   $.get(this._scanningService.runCommand(“SaveToLocal”, param, null));
}
```

6

Scanning Service Code

```
private Stream CommandCallBack(CommandEventArgs args)
{
   if (args.CommandName == “SaveToLocal”)
   {
      try
      {
         // Use constructor of custom class to convert JSON from event args
         SaveToLocalParams saveParams = new SaveToLocalParams(args);
         SaveToLocal(saveParams);
      }
      catch (Exception ex)
      {
         MessageBox.Show(ex.ToString());
      }
   }
   // Handle additional user commands
   return null;
}

private void SaveToLocal(SaveToLocalParams saveParams)
{
   RasterImage page = null;
   RasterCodecs codecs = new RasterCodecs();
   // Check to see how many pages have been scanned
   TwainStatus status = this.ScanningService.GetStatus(
      saveParams.Id, saveParams.UserData);
   for (int pageNumber = 1; pageNumber <= status.PagesCount; pageNumber++)
   {
      // Save the page by appending it to the end of the file
      page = this.ScanningService.GetImage(saveParams.Id, pageNumber);
      if (page != null)
      {
         codecs.Save(page, saveParams.FileName, saveParams.Format,
            0, 1, 1, -1, CodecsSavePageMode.Append);
      }
   }
}
```

9

Conclusion

Scanning documents into digital image formats within a cross-browser HTML5 solution is just one of many real-world solutions you can tackle with LEADTOOLS. Its state-of-the-art Web Scanning SDK makes it possible to create innovative new web applications for capturing documents, and offers a programmer-friendly and modular architecture for adding web scanning to any existing web application or workflow. It also opens the door for extending your web-based document imaging solution with any of the many additional technologies in LEADTOOLS Imaging SDKs. 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: +1(704) 332-5532
sales@leadtools.com

Support: +1(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 25 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