Getting Around the Complexities of Document Scanning and Management via Web Browsers

Web Scanning with Dynamsoft's Dynamic Web TWAIN HTML5 SDK

More and more organizations are turning to document management software to realize their promised advantages. And, the document scanning function itself is arguably the most critical component of a document management solution. If you're a web developer looking to implement a browser-based document management solution, creating the scanning module itself can bring on many troubles. But, there are readily-available alternatives so you can keep trouble at bay.

Document Scanning Background

The document management market continues to blossom. TechNavio research firm forecasts nearly 14 percent CAGR through 2018. There are many reasons for this growth. Organizations are looking to leverage document management solutions for improvements to efficiency, workflow, security, productivity, and of course for potential organizational cost savings.

Document scanning is an essential part of most document management systems. For web developers looking to implement document management, it's a challenge to interact with locally connected devices from web browsers. Usually you need to create browser plug-ins to scan. And there are plenty of browsers to probably need to support. This can cost a lot of time and effort and in the end, often the results are not satisfying. Creating a document scanning module can mean lots of trouble.

What's the Trouble?

When one is looking to develop a solution in-house, it's obviously paramount to consider all the tasks that will be required to complete the project. However, you also need to consider what might be required after the project is completed. This part may bite you as much as the project itself.

The TWAIN 1.9 specification alone is 552 pages long – TWAIN is by far the dominant communications protocol standard for interfacing devices and software. In other words, you must support it in any document scanning module. It could take months just to get familiar with the specification and even longer to then build a TWAIN plugin from scratch. Plus, you will need different solutions to interact with scanners from different environments. For example, you would need an ActiveX control for Internet Explorer, a plug-in for Chrome, Firefox, Safari, etc. The behavior of a browser on different platforms like Windows and Mac OS X are also different and must be considered.

Furthermore, browser providers continually update their applications. This is the part that happens after you think you finished your project. Often, these browser updates require you to do the same to your scanning module to keep up. It takes a lot of effort to stay ahead of browser updates. So, developing a

document scanning solution is hardly the end of the project. You will need to maintain the solution regularly to adapt to application changes or risk suffering down time or user problems.

To remedy all these complexities, Dynamsoft provides *Dynamic Web TWAIN*, which is a document imaging SDK specifically designed for web applications. With just a few lines of code, it enables you to implement a TWAIN-based scanning module in your web-based document management solution.

Intro: Dynamic Web TWAIN Web Scanning SDK

Dynamic Web TWAIN is a cross-browser document scanning SDK. It supports all mainstream browsers including IE, Chrome, Firefox and Safari. And it supports both Windows and Mac OS X. Dynamic Web TWAIN provides simple APIs to enable you to interact with the TWAIN drivers of scanners in JavaScript.

To achieve cross-browser cross-platform web scanning, Dynamic Web TWAIN has multiple options. There are ActiveX and Plugin options for working with old versions of browsers. For modern browsers, there is also a HTML5 option.

As we mentioned earlier, browsers change quickly. Google Chrome ceased support of NPAPI plugins in September 2015. Firefox is dropping NPAPI plugins by the end of 2016 as well. The new Microsoft Edge browser disabled support of ActiveX control. With these newer browser versions there is a need for a new solution for web browser interactions with local TWAIN scanners. This API solution has to work with the modern browsers and be extendable for different browsers and different platforms.

For this purpose, Dynamsoft developed its *Dynamic Web TWAIN HTML5 SDK*.

Dynamic Web TWAIN HTML5 SDK Architecture

The Dynamic Web TWAIN HTML5 SDK includes two main modules – a JavaScript (JS) client and a scanning Windows service. The JS client and service uses a HTML5 WebSocket connection. The Dynamic Web TWAIN HTML5 SDK supports all browsers that are compatible with HTML5 WebSocket on Windows

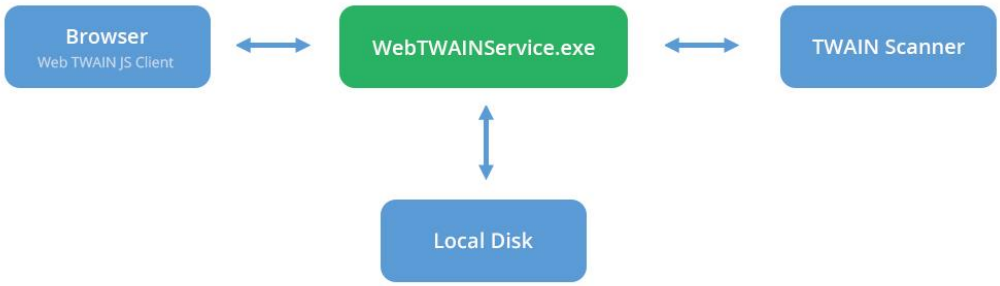
and Mac OS X. Currently, the browser list includes: Internet Explorer 10 & 11; Edge, Chrome v27+ and Firefox v27+ on Windows; and Safari v7+, Chrome v27+, and Firefox v27+ on Mac OS X.

WebSocket is a protocol providing full-duplex communications channels over a single TCP connection. Via the WebSocket connection, a web browser and a server can have a persistent communication sending and receiving data. As such, Dynamsoft leveraged WebSocket to ensure Dynamic Web TWAIN works with Chrome versions that no longer support NPAPI.

If you're interested in more detailed information about WebSocket, you can later refer to Introducing WebSockets: Bringing Sockets to the Web. Moving forward, let's look under the hood of Dynamic Web TWAIN.

HTML5/JavaScript Client

Below is an illustration of the Dynamic Web TWAIN JavaScript library. It accepts operation requests from browsers and sends them to the Web TWAIN Windows service.



For example, you can use a few lines of JavaScript code, such as below, to accomplish scanning functionality:

```
function AcquireImage() {
    if (DWObject) {
        DWObject.SelectSource();
        DWObject.IfDisableSourceAfterAcquire = true;
        DWObject.AcquireImage();
    }
}
```

Self-Host Web Service

At the core of the Dynamic Web TWAIN web scanning solution, there is a “WebTWAINService.exe” which is a self-host Web service.

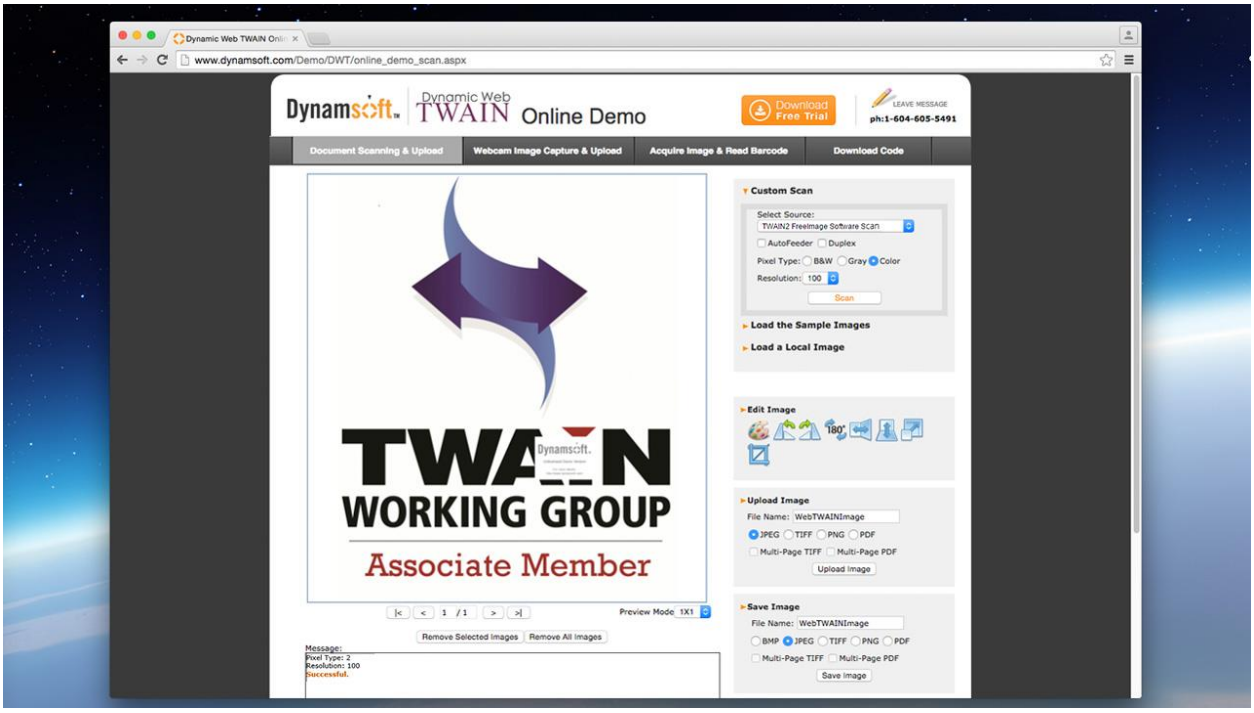
This module accepts user operation commands from the Web TWAIN JavaScript client. It interacts with local TWAIN drivers. If you have requirements like loading existing images or saving images from/to local disk, the module will perform the file operations in a separate process.

```
void Process_Request(){
  JsonCommand cmd;
  cmd.parse(request);

  TwainClient & client = TwainClient::GetInstance();
  client.DistributeCommand(cmd);
}

TwainClient::TwainClient(){
  m_functions.add("RotateLeft", RotateLeft);
  m_functions.add("LoadImage", LoadImage);
}


void TwainClient::RotateLeft(int iImageIndex, BOOL * retVal){
  HANDLE hDIB = GetImageFromIndex(iImageIndex);
  CImage image(hDIB);
  *retVal = image.RotateLeft();
}


void TwainClient::LoadImage(char * pszFilename, BOOL * retVal){
  SOperationParam stParam;
  stParam.actionType = Plugin::LoadFile;
  stParam.pszLocalFile = pszFilename;

  *retVal = FALSE;

  // send params to pipe server
  if (DoOperationInProcess(stParam)) {
    DWORD dwSize = 0;

    // find size & bytes return from pipe server
    HANDLE hMem = FindLoadedFileMem(pszFilename, &dwSize);
    if (NULL == hMem) {
      return;
    }

      *retVal = LoadImageFromBytes(dwSize, hMem);
    GlobalFree(hMem);
  }
}
```

A Demo Application

Below is a link to a demo application built using Dynamic Web TWAIN. With the demo, you can scan documents from TWAIN devices, edit the images and save the images to local /server disk, database or document libraries. The capabilities you will see in the demo are just some of the many functions you can more easily implement when using Dynamic Web TWAIN.

Dynamic Web TWAIN Demo working in Edge browser on Windows



Dynamic Web TWAIN Demo working in Safari on Mac OS X

[Try out the online demo >]

Conclusion

The task of developing a web-based document scanning solution is a challenge. It's made even harder when you understand the daunting task of having to keep up with browser updates. But, with the right tool – such as a HTML5-based TWAIN scanning SDK like Dynamic Web TWAIN – you can deliver a document scanning module quickly. You can then focus your time and resources on better satisfying your customer requirements.