Embracing HTML5

A Piece of the Document Viewing Puzzle
Embracing HTML5: A Piece of the Document Viewing Puzzle

For businesses and organizations across the globe, being able to access and process documents across a wide user base with minimal effort and technology is essential to success. One attractive approach to solve these challenges is to use web-based methods to most easily reach target audiences. However, implementation can be difficult with people using a variety of computing platforms and technologies. As a result, more and more companies are turning to HTML5 technology in 2013 as a solution to these document viewing and processing challenges.

HTML5 technology has quickly become an attractive option for companies offering or creating document viewers for a variety of reasons. Most importantly, the technology, which is present in all of the latest browsers, does not require any application downloads, plug-ins, or an installation process.

But despite these obvious benefits, many corporate users are still concerned about potential trade-offs when using this technology. Namely, there are concerns about users having the updated browsers necessary to view HTML5-dependent content. For example, Internet Explorer versions 6-8 do not support HTML5. In March 2011, Forrester reported that 60% of all corporate desktops were PCs running Windows XP. Of course, in the hands of skilled document viewer developers, this is not an issue. When a viewer is supplemented with some other important technologies on the back end (CSS, JavaScript, and AJAX), developers can combine the advantages of HTML5 with the aforementioned technologies to create a sophisticated document viewer that can work on any browser with no installation.

In order to further assuage some of these concerns, we will look closely at HTML5 technology as it relates to document viewers. What follows is a description of HTML5, its competition, its limitations, and how those limitations can be addressed when paired with other technologies to create a sophisticated document viewer...
What is HTML5?

It is important to remember that HTML (hyper text markup language), which is a grammar used for structuring and presenting content for the World Wide Web, has been around since the early 1990s. HTML5 is not an entirely new technology, but rather the in-development standard for the fifth major version of HTML.

The main goal of HTML5 is to improve the current HTML language so it can more easily support the latest multimedia content and overcome the fragmentation issues that the web industry has consistently battled.

HTML5 offers many new features and upgrades:

• Detailed processing models to encourage more interoperable implementations
• Extends, improves, and rationalizes markup available for documents
• Includes the important canvas command needed for displaying and manipulating images within a webpage
• Introduces markup and Application Programming Interfaces (APIs) for complex web applications
• Removes the need for proprietary plug-ins or installations

As of January 2013, HTML5 is still being developed and implemented as a standard. The World Wide Web Consortium (W3C) and publishers of HTML5 are still making updates and additions. In December, the W3C released a new draft of specifications referred to as HTML5.1. This initial release, which included improvements to video captioning, image accessibility, and iframes, is another step toward the W3C establishing a concrete, formal HTML standard. For the time being (and for practical purposes), many HTML5 features are already in place and being supported in browsers. As more of these new HTML5 updates and releases arrive, document viewer developers will need to pay close attention and continue to tweak their software accordingly to ensure full compatibility.

Looking to the future, HTML5 is positioned to become a cross-platform solution that will bring us closer to a universal computing experience once it is completely standardized. But for the time being, it must be supplemented with additional technologies in order for the software utilizing it to reach the most users possible.

Name: HTML5
Years: 1995 - ???
Family: HTML
Ownership Group: World Wide Web Consortium (W3C)
Description: 5th major version of the hyper text markup language used for structuring and presenting content on the world wide web

Features/Stats:
• Removes the need for proprietary plug-ins or installations
• Detailed processing models
• Extends and improves markup available for documents
• Important canvas command
• Markup and APIs for complex web applications
A leg up on Adobe Flash-based viewers

Although more and more document viewers are turning to HTML5 technology, there are still some Adobe Flash-based, as well as ActiveX-based (a software component of Microsoft Windows), document viewers on the market. Flash is an attractive multimedia platform because it provides a rich interactive experience for users and also has great tools for designers and developers. It is best known for combining multi-media and interactive animation elements (audio, video, hand-drawn animations, warping effects, and clickable elements) into Flash objects (independent, self-contained objects).

Flash can be seen all across the web in data-driven applications, gaming, and rich media. In the gaming world, Flash allows developers to develop high-performance, console-quality games in 2D and 3D for desktop, mobile, and TV audiences. Similarly, Flash assists developers in delivering secure and consistent HD-quality videos across multiple devices, desktops and TVs.

Despite these advantages, Flash also carries with it some significant drawbacks. First and foremost, it requires a download and installation. Many corporate users have a “no install” goal to minimize customer support. Second, Adobe Flash, due to security concerns, frequently requires downloads and updates to keep up with code fixes, patches, and security issues. This poses a significant obstacle when trying to reach the widest amount of users as possible.
On top of the download requirement, Adobe Flash has never been supported by Apple on mobile devices and tablets because of the proprietary nature of the technology and issues with reliability, security and performance. This is obviously a major disadvantage with so many internet users currently relying on Apple products such iPhones and iPads for their browsing needs:

- As of February 2012, Nielson Mobile Insights revealed that iPhone (iOS) users represented nearly 32% of the cell phone market.

- iPhone users also use their web browser much more than their counterparts on other devices by significant margins. According to NetMarketShare, the iPhone has almost a 65% usage, leaving all other web browsing devices far behind (Android is the next closest at 9%).

- At the end of Q3 in 2012, Apple had a 55% share of the worldwide tablet market.

- Compounding the issue further, Adobe Flash is no longer being supported on Android versions 4.1 and onward, severely limiting its mobile reach going forward in 2013.

So even though Adobe estimates that over 96% of web browsers have Flash installed, once you adjust the statistics to include mobile and tablet factors for iOS and Android 4.1 users, a significant amount of users will not be able to view the Flash content. When coupled with the time-consuming download and installation steps, Apple’s firm position (as well as Android’s moving forward) clearly illustrate why Flash isn’t a good choice for a document viewing application despite the rich interactive capabilities it offers.
The limitations of HTML5 for document viewers

As the core component to a pure HTML document viewer, HTML5 technology brings significant benefits to the table, such as the important canvas command, requiring no installation, and allowing for true cross-platform support. However, it is not the only component to a sophisticated document viewer. Below are some of the limitations of HTML5 and the challenges they pose for document viewers:

- **Browser compatibility issues**: The main cause for concern surrounding HTML5 is that there are many computers out there that are still using Windows XP (and sometimes even Vista and Windows 7) and utilizing old versions of Internet Explorer (6, 7, and 8). This is particularly true for large companies, as the number of systems with these older browsers is counted in the millions. These older browsers use variants of either HTML4 or XHTML 1.0 and don’t include the special capabilities defined in HTML5. Even then, Internet Explorer 6-8’s support for existing standards like HTML4, XHTML1, and CSS1-3 is incomplete and poses problems. Regardless of using HTML5, IE 6-8 presents both functionality and appearance issues if special attention is not taken.

- **Limited file support**: HTML in general (5 or otherwise) doesn’t natively support any file formats. Individual browsers are responsible for rendering or supporting whatever image formats are downloaded, as there is no required or suggested list of formats prescribed by the HTML standard. Most browsers support JPEG, PNG, BMP, and GIF files and Google Chrome comes with a built-in PDF viewer. But to display other file types and formats (such as TIFFs and Word documents), a plug-in or conversion on the server side is necessary.

- **Limited functionality on its own**: HTML5 doesn’t automatically provide sophisticated functionality for image manipulations inside of a browser like annotations, search functions or thumbnails.
Solutions to these obstacles: Creating a sophisticated viewer with dynamic capabilities

There are many things that must occur on the back end of a document viewer before customers can enjoy a sophisticated, zero footprint HTML viewer that works on any browser. Skilled developers must pair the HTML5 technology with a couple of other technologies and server-side components to give the viewer interactive capabilities.

Special coding solves browser compatibility issues: Older versions (6-8) of Internet Explorer have broken implementations of HTML. Specific to HTML5 is the important graphics drawing command, canvas, which is needed for displaying and manipulating images. To solve these IE issues, developers can use polyfills written in Javascript and some specific CSS files to make sure the canvas-dependent elements function and display properly in these older versions of Internet Explorer.

A content server expands file support: Because browsers utilizing HTML5 can’t support more than a few different file types, a content server is a necessary component to a document viewer with robust viewing capabilities. With a content server (which in the right hands can be written to support varying platforms and applications, including Java and .NET), a variety of file formats such as TIFFs and Word documents can be retrieved and automatically converted on-the-fly from their native format into a temporary image (PNG) that will display in the browser.

Additional technologies give the document viewer sophisticated capabilities: Along with using HTML5 functions, a modern document viewer takes advantage of JavaScript and CSS, and utilizes AJAX technology to make calls to the content server.

Solution: Developers can use polyfills written in Javascript and specific CSS files to make sure the canvas-dependent elements function and display properly in versions 6-8 of Internet Explorer.

Solution: A content server expands file support to retrieve and convert a variety of file formats including TIFFs and Word documents to a temporary image (PNG) that will display in the browser.

Solution: Additional technologies (JavaScript, CSS, and AJAX) give the document viewer sophisticated capabilities.
JavaScript (open source client-side scripting language): JavaScript is the client-side scripting language used primarily in webpage creation and rendering for enhanced user interfaces and dynamic websites. Almost everything in a document viewer requires Javascript code for implementation.

CSS (Cascading Style Sheets): CSS is how the layout of a web page is described in a browser. Developers can use different sets of CSS rules for different webpages, as well as different platforms (desktop, tablet, mobile). As mentioned above, including some specific CSS files allows the document viewer to display properly in IE 6-8.

AJAX (asynchronous JavaScript and XML): AJAX is a programming paradigm that refers to a set of advanced browser technologies. AJAX allows for the retrieval and sending of data to a server without a page reload. With AJAX technology, the document viewer operates more quickly and more efficiently as it communicates with the content server, making for a better user experience. These AJAX calls and communications with the content server allow for the retrieval and saving of annotation and redaction changes within the document viewer.
The takeaway: Document viewers in a brave new HTML5 world

Even though HTML5 is still being developed as a standard, there is no downside to relying on HTML5 technology for document viewers now—provided it is in the hands of skilled developers who are pairing it with other necessary technologies.

Creating a sophisticated, browser-based document viewer like the one described above is not easy. Along with the existing technologies, it takes plenty of time, special coding, and configuration on the back end before it can become a universal document solution that supports all of an organization’s requirements. But with effort, skill, and time, a zero footprint document viewer that works on any browser and for any document can be a reality.

In the end, HTML5, along with JavaScript, CSS, and AJAX, are all just technologies. HTML5 is a large part of it, but it is only one piece of the puzzle. In the right hands, all of these technologies can be engineered to form a pure HTML document viewer that enables organizations to effectively and efficiently tackle all of their document viewing and processing demands.