Goodbye Applets, Hello HTML5
Introduction

A lot of things that were popular and relevant in the 1990s have since become obsolete—think floppy discs, Discmans, and VCRs—or are quickly becoming obsolete (looking at you, fax machines, land lines). And soon to be joining their ranks: browser plug-ins and, more specifically, Java applets. With the news that Oracle plans to discontinue its Java browser plug-in—first introduced in 1995—in its next big release (JDK9), it appears that yet another formerly hot technology is on its way out.

The Java programming language isn’t going anywhere, however, and neither is JavaScript, a completely different language. The only thing that is being sent out to pasture is the JRE plug-in that allows Java applets in the browser. Due to security concerns, browsers have stopped supporting them and the browser’s own HTML5 technology has made them replaceable.

In this eBook, we’ll take a look at the history of applets and plug-in technology, why they are being phased out, and how document viewing technology is moving on accordingly.
Netscape 1.0 is released.

The Java programming language and Java applet is introduced by Sun to provide cross-platform operations. The applet can be embedded into a webpage and provide capabilities such as rollover effects, games, visualizations, and other basic interactivity.

The HotJava browser is first released—a modular, extensible web browser from Sun Microsystems and the first browser to accommodate Java applets.

Microsoft’s Internet Explorer 6 is released (around the same time Windows XP is originally released). Because these early browsers are limited in their functionality, browser plug-ins are needed to add features.

Firefox 1.0 is released as an offshoot of the older Mozilla browser.

The heyday of Java applets. During this time, applets are everywhere. According to Motherboard.Vice.com, they are “powering early camera-phones and covert CCTV set-ups, visualizing protein sequences, processing and analyzing shipping data, enabling early online learning tools and 3d-enabled cyber workspaces, and more”—all in addition to what they were doing within “the context of everyday web interaction.”
2006
Microsoft Internet Explorer 7 is released (five years after IE6), browser development stagnates and Microsoft pulls its developers off IE and stops developing it entirely.

2007
Microsoft releases Silverlight to provide streaming media and animation as a competitor to Flash.

2008
Chrome 1.0 is released by Google.

2009
IE8 and IE9 are released with not many improvements over the initial IE6.

2011-2014
Everything changes with the release of Google's V8 JavaScript engine, which means JS code can now be compiled by Chrome and run on the client machine. Google, Mozilla and Apple race forward with rapid advances in browser technology. Meanwhile, security vulnerabilities in Flash, Silverlight, ActiveX and Java applets are exploited by malware hackers.

2015-2016
The official demise of most plug-ins. Browser vendors begin removing plug-in support, eliminating the ability to embed plug-in based technologies. Oracle announces plans to discontinue the Java browser plug-in across all browsers in its upcoming release.
What’s The Deal with Plug-Ins?

While Snowbound Software has been a champion of Java applications on both servers and clients for almost 20 years, the popularity of plug-ins on end-user systems has resulted in many security concerns. They have been on the receiving end of many attacks targeting vulnerable client systems from many angles, including ActiveX, Flash, Silverlight and Java.

As attacks have become more sophisticated by taking advantage of the very redeeming capabilities of these applications, it has become increasingly difficult to keep evolving these older technologies.

At this time, most browsers have either discontinued support for these web-related applications or they are rapidly working to do so. When Oracle announced that they would discontinue support for Java applets, it was clear that the burden of keeping Java applets safe had become uneconomical with the liabilities increasing. It should be noted that Oracle/Sun will continue to provide Java environments for servers (after all, they still sell the very popular Sun Unix servers that run Oracle and many other enterprise applications), but the dedication to applets is terminating.

Businesses that have resisted moving from applets and older browsers have come to realize that being cut-off from the web in order to secure their client systems has diminishing returns and results in productivity losses. It is also possible that even in an enclosed, Internet-free environment older browsers are still vulnerable in the event of an internal attack or breach.

Many companies and application suppliers have years—or even decades—of experience and practice invested in Java applets, so change will not come quickly. But the writing is on the wall that it is time to start, or at the very least, speed along current efforts.
**7 Reasons Plug-Ins Are Being Phased Out**

1. **Security:**
   The plug-ins used by browsers are often less secure than the browsers themselves. Java and Flash have been victims of some of the largest security threats on the web in the last few years. The fact that these plug-ins are the same for all browsers (regardless of operating system) means the threats make all operating systems equally vulnerable to the same attack.

2. **No Sandboxing:**
   Sandboxing is a security technique that restricts applications from gaining access to the important parts of your computer, thus isolating the malfunctioning application or malicious attacks that may channel through it. Most plug-ins like ActiveX do not run in this manner and thereby allow potentially harmful threats to enter the operating system.

3. **Cross-Platform Problems:**
   Plug-ins are developed by a single company, so it is not possible to have true cross-platform support as the developing company only supports the plug-in on their platform. For example, Flash does not run on iOS devices and ActiveX does not run natively on browsers other than Internet Explorer. HTML5 is an open standard that allows the applications to run across multiple platforms.

4. **Stability:**
   Plug-ins themselves are a major contributor to browser crashes. Plug-ins provide the support for some of these applications to run in the browser so the applications themselves become slaves to the willingness and competency of the plug-ins.

5. **Flash:**
   Flash has been almost completely phased out of all modern applications, as it requires frequent updating to keep it secure and is no longer needed to support the many different tasks that older browsers depended on it for.

6. **Java:**
   As Java plug-ins have increasingly become the leading cause of attacks, their security has become impractical to manage.

7. **ActiveX:**
   ActiveX is Microsoft’s version of a plug-in. It was created to allow important functionality (like document viewing) in the browser. HTML5 development has made these functions available without the use of ActiveX. Microsoft’s ActiveX is a microcosm of why browser plug-ins are seeing the end of their usefulness—security issues, outdated support, limited to Internet Explorer, and general instability.

“Applets are a technology that are very much of their time, and they have not aged well. The technology proved to be very difficult to evolve, and so applets have not been considered to be a modern development platform for many years now. However, they doggedly persist due to some large early adopters being very resistant to change. The resulting effect to the ecosystem is that Java applets are still present in the platform, and are a major contributor to security problems.”

- Benjamin Evans, in his book *In Java: The Legend*
It should be clear by this point that web applications that don’t require plug-ins are the future. HTML5-based web applications are currently the best solution for web client applications. One such application that has benefitted greatly in this shift to pure HTML technology is document viewers. For those in need of a powerful solution to viewing hundreds of different documents from any browser or device, HTML5 document viewing is the answer.

- They work on any system that offers a modern browser, including desktops, laptops, tablets and smartphones.
- They minimize client support dramatically because there is no environmental requirement (other than a browser).
- Product updates occur automatically and invisibly as soon as the user logs in.
- Security is improved because browser and OS security enhancements come out regularly.
- The removal of browser plug-in applications like Flash, ActiveX and Java eliminates the most common security vulnerabilities.
- The HTML, Javascript and CSS ecosystem is dynamic and inventive with advances for additional capabilities coming out regularly.
However, it is important to know that all HTML5 viewers are not created equal. If you are in the market for a pure HTML5 viewer, make sure it easily integrates with your current systems and has the sophisticated features required in your document workflow.

**HTML5 Document Viewer Check List:**

- Responsive user interface for mobile and tablet viewing.
- Powerful API for customizing the app.
- Robust server support—where all the heavy lifting happens.
- Broad format support beyond TIFF or PDF, depending on your document requirements.
- Ability to integrate with other server applications through custom or industry-standard connectors.
- SVG capability for crisp rendition of all text-based and CAD documents at any zoom level.
- Workflow features such as annotations, thumbnails, document tabs, document/page manipulations, virtual documents, and redactions.
- A vendor with experience working with demanding large corporations.
- A vendor that can guarantee the regulatory compliance required by financial institutions and other industries for data security, background checks, disaster recovery protection, insurance coverage, IP indemnification, and more.
- In-house experts to advise and support the customer’s special requirements.
- Java and .NET server platform support to meet specific environmental needs.
- Wide-ranging partnerships with system integrators who are experts across a variety of businesses—records management, content management, claims processing, loan processing and mortgage processing.
Conclusion

With the initial benefits that plug-ins brought to the table now rolled into the browsers themselves, there is no longer a reason for plug-ins to exist. They served a very useful purpose in the early web era—turning web pages into feature-rich computer programs—but technology has progressed rapidly since then and realized those benefits through other means.

Modern browsers, and more specifically, HTML5 technology, have ushered in a new era of more secure and powerful, browser-based applications. Document viewers are just one example of an application that has benefited greatly from pure HTML technology. Browser-based document viewers that provide users with access to hundreds of different document types from any browser or device have proved to be the perfect replacement for previous viewers dependent on applets and plug-ins. Just make sure you do your research when deciding on an HTML5 viewing vendor, as all viewers are not created equal.