



# Flex & AJAX

## Friends or Foes?

Anthony Franco  
President EffectiveUI  
Updated June 18th, 2007

## Introduction

Many developers will tell you that Flex and Ajax are apples and oranges; compliments to one another that aren't always in direct competition. While this is certainly true, in the space of rich Internet applications (RIAs), there are still many situations where both technologies provide similar solutions to the same problems. The purpose of this white paper is to provide a comparison between Flex and AJAX in those situations where they are both relevant to the problem at hand.

### What Is Flex?

Flex is a free, open-source platform and component library for developing applications that can be deployed on the ubiquitous Flash Player. When you think of Flash, you may think of the Player or the Flash development IDE. Flex is just another way of creating applications to be deployed on the Flash Player. Flex is specifically geared toward building robust, rich Internet applications -- internet applications that behave with the same interactivity and engagement as desktop applications. There are two languages in the Flex Framework MXML and ActionScript. ActionScript is a strongly typed ECMA compliant scripting language and MXML is typically used for layout, styling and data binding.

### What Is AJAX?

AJAX is an acronym for 'Asynchronous JavaScript And XML'. It combines coding with HTML, JavaScript, and XML to allow interaction between the client and server to occur 'asynchronously'. The 'asynchronous' part is important - it means that user interaction with the interface isn't broken up by page refreshes every time the server has something to do. In essence, AJAX is a series of techniques and tricks to make traditional web pages more interactive.

## Feature Comparison Matrix

Caveat: AJAX services / tools / libraries / compatibilities and components are morphing everyday. This matrix is designed to compare the most important features of the most robust AJAX frameworks with the Flex framework. It is not meant to be a comprehensive list, and is subject to some interpretation. In addition, since Flex and Flash are irrevocably bound together, many of these comparisons do not make the distinction between Flex, Flash or the Flash Player.

	Why Care?	AJAX	Flex / Flash
<b>Animation</b>	Animation support allows developers to indicate state change, orientate the user on navigational changes, or to simply entertain & immerse the user.	Limited support and difficult to author anything but simple linear animations	Richly supported
<b>Bitmap Manipulation</b>	1) Allows users to alter bitmap imagery real-time, client side 2) Gives developers flexibility to create image effects, distortions and alterations at runtime 3) Aids in animation support (think motion blurs and distortion effects)	Supported inconsistently in Opera, Firefox and Safari. Can also be achieved through extensive server-side processing and client-server interaction	Supported natively
<b>HTML Rendering</b>	In some circumstances it is beneficial to render HTML within your application. For example, users might generate complex HTML content for other users to see.	Fully supported	Limited support (no HTML tables, no JavaScript, no frames, etc)
<b>Video and Audio Streaming</b>	Video is the hottest thing on the web right now. You may not think that it matters to you, but keep in mind that most companies decide to augment an application later with some type of video enhancements like video training, corporate communications, or entertainment. Now, more than ever, video is becoming a big part of online communication.	No native support (user needs a separate plugin like Quicktime, Windows Media Player, or the Flash Plugin to play video in AJAX deployments)	Supported, video quality is pretty good but needs to improve before widely adopted by media companies. Video and audio capture through user's installed devices is also supported.
<b>Development Environments</b>	The better the development environment, the easier it is to create code quickly and the more developers will adopt the technology.	Free - Google Toolkit Free - Echo 2 Free - jsLINB Free - Rico \$5400 Zapatec (many more)	3 Authoring environments: \$500 Flex Builder (Eclipse Based Plugin) \$700 Flash CS3 Free - Any Text Editor
<b>Runtime</b>	The runtime is what interprets the code into executable actions to the processor. Inconsistencies, bugs and performance issues in the runtime can seriously degrade the user experience.	Each browser is required to interpret code individually.	A single browser plugin provides a single cross browser cross platform runtime
<b>Strongly Typed / OOP Language</b>	Strongly Typed and Object Oriented Programming methodologies make it easier to create code with larger teams by enforcing certain coding patterns, and metaphors across an entire code-base	Supported in some frameworks, but still compiles down to a prototype structure. Most libraries do not mix well with one another and can commonly cause conflicts.	E4X Compliant

	Why Care?	AJAX	Flex / Flash
<b>Vector Graphics</b>	Vector graphics give designers the ability to create vector based skins and graphics for their applications, making the footprint of the application dramatically smaller. This also gives developers the ability to create complex data visualization components that can dynamically update, animate and redraw without the need for server round-trips.	Supported through VML in newer releases of IE and limited support with SVG in some of the other browsers	Supported natively
<b>Security</b>	How easy is it to accidentally or nefariously create an application with security holes that threaten your servers or the user's machine?	Both AJAX and Flex utilize the browser's security sandbox to ensure the security of the desktop. Hackers may have an easier time finding security holes in your systems by peering into the source code of your AJAX application while the code geared for the Flash Player is obscured by compilation and can be encrypted.	
<b>DOM Control</b>	Accessing the Document Object Model allows an application developer to control the browser itself.	Supported	No native support. Must use external calls to JavaScript
<b>Server Communication</b>	Obviously, the ability to connect your application to a services is what building an online application is all about.	Both AJAX and Flex allow you to communicate using web services, REST, and simple XML. Binary sockets are supported with the Flash Player giving you the ability to communicate very light weight data packets bidirectionally (pull and push)	
<b>Search Engine Optimization (SEO)</b>	Sometimes, when dealing with RIA technologies SEO is irrelevant. Search engines operate in page based metaphors and some RIAs are dynamic and interactive such that SEO is not appropriate.	Friendlier SEO by nature as the code is natively HTML. However, most search engines will not follow JavaScript links. AJAX needs to be very carefully architected if SEO is critical.	Limited support. SEO is achieved through meta tagging and alternate content publishing.
<b>Accessibility</b>	If you applications are targeted for large organizations or government agencies, accessibility is a core critical requirement. Also, isn't it just the right thing to do?	Limited, few frameworks claim to have support	508 compliant
<b>Web Analytics</b>	Measurement is the key to understanding the success of your application. It also allows you to understand where you can make incremental improvements to optimize the user experience and help accomplish business goals.	RIA analytics tools are woefully lagging behind. Omiture, Web trends and Google Analytics all have limited support and poor RIA reporting. EffectiveUI has a set of homegrown tools to properly measure and report user behavior within an RIA.	

	Why Care?	AJAX	Flex / Flash
<b>Automated Testing</b>	Large scale, enterprise applications require the ability to test user interactions in an automated way in order to increase testing accuracy while reducing the budget and schedule of quality assurance cycles.	Many web-page automated testing tools will work fairly seamlessly with simple AJAX implementations.	Flex components include an automated testing framework that tie into tools like QTP or home-grown solutions.
<b>Openness</b>	How “locked in” to a proprietary technology are you when choosing the right platform?	There are many different licensing options for the various AJAX frameworks. Anywhere from open source to commercially licensed.	Flex and ActionScript are open source. the Flash Virtual Machine (FVM) is not. Adobe has decided to keep the runtime closed to control code versioning, standardization and security.
<b>Extensibility</b>	Enables 3rd parties to create tools, plugins, components, etc.	Since AJAX is an extension of HTML and JavaScript, the framework is inherently extensible	The Flex components can be easily extended, and new ones written. The Eclipse based Flex Builder is also significantly extensible.

*“AJAX is an incremental path to a better user experience [however] the AJAX path itself forks. Just because a development team has chosen to go with Ajax doesn’t mean that its tool selection effort is complete. There remains the important decision of how many and what type of Ajax frameworks to use.*

*Flex delivers productivity in the long run but takes more time to learn. Adobe is an attractive RIA solution because it combines powerful development tools (Flex) with a near-universal browser plug-in (Flash) that’s designed to simplify RIA deployment. Managers who have concerns about small commercial Ajax providers or support issues with open source Ajax frameworks consider Flex a safe alternative. But there is a downside: To be productive in Flex, developers have to learn two new languages and an extensive runtime framework — something that can take between one and three months to learn.”*

*Jeffrey Hammond, Forrester Research. “AJAX or Flex, How to Select RIA Technologies”*

## Compatibility

Right now, there is a heated debate on which platform is more compatible than the other. Simply put, JavaScript and DHTML are supported on more desktops than the Flash Player.

Unfortunately, its not that simple. The differences in how browsers interpret AJAX code is where this discussion gets very complicated, very quickly. AJAX frameworks attempt to obscure browser differences by creating components and libraries that account for browser and operating system inconsistencies. However, these frameworks do not publish ubiquity statistics, making it hard (if not impossible) to make educated compatibility decisions on which framework(s) to choose. In addition, if you need to customize a component, create a new control, or extend a library, you now have to worry about your code being cross-browser compliant. As the sophistication of your AJAX application increases, the potential compatibility issues will increase exponentially.

Flex, on the other hand, uses a ubiquitous runtime, this means that the Flash Player uses the same virtual machine to run your code on every browser, therefore mitigating potential compatibility issues. You should never have to rely on browser hacks or exception coding based on operating systems. In addition, Adobe publishes their Flash Player compatibility statistics and updates those statistics four times a year. Decision makers tend to trust technology that is constantly measured on important metrics, but these statistics are often used against Adobe to point out that the runtime is not as compatible as JavaScript. However, remember that JavaScript runs differently on each browser and that **your** code may not be as compatible.

Developing and testing large applications across browsers and operating systems dramatically increases production costs. This is the primary reason why many firms use AJAX to make incremental improvements and modifications to their online applications, and use Flex for large scale deployments where performance, scalability and user experience are the highest priorities.

# Performance

The most definitive performance comparison between the 2 platforms was done by James Ward in April, 2007:



*"While these results may be eye-opening to some, once you understand some fundamental differences between Flex and Ajax, they end up making a lot of sense. Flex applications run within the Mozilla Tamarin VM inside Flash 9. Tamarin does JIT compilation which makes code execution 10 to 1000 times faster than interpreted JavaScript in the browser. Also Tamarin can be very efficient when dealing with typed objects.*

*In addition, Flex supports a compact binary object serialization protocol, called AMF3, which is extremely fast. There are numerous Open Source implementations using various back-end technologies for AMF3, as well as a commercial / supported Adobe product, called Flex Data Services. If you control both sides of the wire in your application, than there is rarely a good reason to serialize and de-serialize text. Objects are blazing fast as you can tell by this benchmark. AMF3 is also typically much more efficient across the wire (even without compression) and consumes much less client side memory. JIT compilation and binary object serialization are the primary reasons why Flex AMF3 is so fast, even in the test with the full 20,000 rows. And, it's not just faster for loading -- it also speeds client side sorting and filtering."*

**James Ward "Ajax and Flex Data Loading Benchmarks"**

## Interoperability

In some cases, the answer is not "either Flex *or* AJAX", but "both Flex *and* AJAX". Each have their place and can also work together quite smoothly as the needs of the application dictate. For example, you can leverage the rich animation, vector drawing and layout management of Flex, and incorporate an AJAX layer to allow your application to render complex HTML. More interesting applications can make use of Flex/Flash and AJAX to support each other. Using simple API calls to and from the Flash player to JavaScript allows simple interaction between the two and Adobe's Flex/Ajax bridge ([http://labs.adobe.com/wiki/index.php/Flex\\_Framework:FABridge](http://labs.adobe.com/wiki/index.php/Flex_Framework:FABridge)) makes more complex interaction possible with a minimum of effort. Some great applications already use this type of approach today. For example:

- Earth Measurements (<http://www.earthmeasurements.com/>) is a great example of an augmentation of an already great rich application. Flex overlays work with the underlying Google maps layer to let you measure distance between points on a Google map. A great example of complex, tight, integration between the two technologies.
- Google Finance (<http://finance.google.com>) is a great application for checking out how a certain company is doing financially. For instance, have a look at Microsoft at <http://finance.google.com/finance?q=MSFT>. Notice how elegant and simple this stock's performance graph is. It's easy to see the stock's value at different levels of detail with zooming and tool tips - all things that are easy to do in Flex/Flash, and difficult if not prohibitively difficult to do in AJAX. But Google also maps landmarks in the stock's performance to news about the company that may shed light on why the value is taking turns up or down. The news is just text and links, no reason not to use HTML if you already have the expertise and the application makes no more technical demands. The power of Flex/Flash and AJAX interoperability shows through when you click a milestone on the graph or in the news list and the graph and news communicate to highlight a selection from the other. Also, scrolling the graph keeps the news in sync. A great example of each technology in it's place and working together for a great application.

## Practically Speaking

You're probably saying to yourself, "Okay, all this information is great. But bottom line it for me. What technology should I use?". The answer is that it depends (big help, right?) As with any software development process, you need to figure out which technology best solves your specific problems. It may be helpful to talk about what types of applications are suited for each:

Use AJAX:

- To make incremental usability enhancements to an existing web site.
- For building "widgets" that do not need a team larger than a couple of developers.
- When you have existing, internal JavaScript and HTML expertise.

Example uses could be navigational elements, simple calculators, and client-side validated forms.

Use Flex:

- When you need to develop applications that require a robust, scalable rich Internet application.
- Where you require sophisticated and interactive data visualization.
- When video & audio playback or web camera / microphone capture is a requirement.
- Where you require complex animation or bitmap manipulation.
- When the graphic design is core critical to your business needs.

Example uses could be product configurations, workflow/process/inventory management applications, video conferencing, playback & editing, immersive or entertaining experiences, data visualizations, and management dashboards.

Use Both:

- When SEO and an immersive experience are equally important.
- When neither meet all of your needs (for example: video playback and HTML rendering are both requirements).

Example uses could be WYSIWYG HTML editors, and user generated content portals.

Additionally, in real-world use, Flex typically requires less coding to build the same or better functionality. The learning curve is higher, but the development times are significantly lower. As one of EffectiveUI's senior architects, put it:

*"I really believe that Flex development is faster than AJAX simply because you end up **writing less code**. The Flash runtime just handles a ton of [stuff] for you. I am technology agnostic (or try to be), but one thing I certainly am is **lazy!!!** I'm too old to be motivated by caffeine and late nights. So Flex has my vote in the productivity camp. Flex vs. C# feels to me like the same productivity boost that C# vs. C++ did a few years ago."*



# More Resources

If you would like to dig deeper on this subject, the articles, blogs and papers below have helped us in creating this article:

Ted Patrick, Inside Yahoo! Interview "AJAX vs Flash"

[http://www.onflex.org/ted/2006/09/ajax-vs-flash-inside-yahoo\\_27.php](http://www.onflex.org/ted/2006/09/ajax-vs-flash-inside-yahoo_27.php)

Pete Cashmore, "Flash, AJAX and Yahoo Maps: Does the Technology Matter?"

<http://mashable.com/2005/11/03/flash-ajax-and-yahoo-maps-does-the-technology-matter>

James Ward, "AJAX and Flex Data Loading Benchmarks"

<http://www.jamesward.org/wordpress/2007/04/30/ajax-and-flex-data-loading-benchmarks>

Jeffrey Hammond, Forrester Research, "Ajax Or Flex?: How To Select RIA Technology"

[www.forrester.com/go?docid=40989](http://www.forrester.com/go?docid=40989)

Ryan Stewart, "The Universal Desktop"

<http://blogs.zdnet.com/Stewart/?p=216>

Wikipedia

[http://en.wikipedia.org/wiki/AJAX\\_vs\\_Flex](http://en.wikipedia.org/wiki/AJAX_vs_Flex)

Flex.org

<http://www.flex.org>

Ajax.org

<http://www.ajax.org>

Flash Player Penetration Statistics

[http://www.adobe.com/products/player\\_census/flashplayer/version\\_penetration.html](http://www.adobe.com/products/player_census/flashplayer/version_penetration.html)

AJAX Browser Compatibility Matrix

<http://www.musingsfrommars.org/2006/06/howre-we-doing-now-ajax-browsers.html>

Adobe Flex Product Information

<http://www.adobe.com/products/flex/productinfo/overview>

Mozilla Development Center

<http://developer.mozilla.org/en/docs/AJAX>

Flex and AJAX Bridge

[http://labs.adobe.com/wiki/index.php/Flex\\_Framework:FABridge](http://labs.adobe.com/wiki/index.php/Flex_Framework:FABridge)

The Artemis Project

<http://artemis.effectiveui.com>

EffectiveUI's Website

<http://www.effectiveui.com>

## About the author:

As a President of effectiveUI, Anthony Franco has the unique opportunity to listen and engage with companies like eBay, Adobe, Discovery Networks, Business Objects, Viacom, GE, and NBC to help them strategize how innovative thinking can completely change the way they do business. He tries not to focus too much on the "technology", but on how innovating on the user experience can create breakthrough business results.

## About EffectiveUI

EffectiveUI has emerged as the leading user-centric design and technology services provider. Through their unique combination of user experience strategy, breakthrough design and leading development expertise, EffectiveUI helps clients by maximizing customer engagement and their online business opportunities. They are known for providing highly engaging, immersive, and effective rich Internet, desktop and mobile applications. Some of the most prestigious and reputable brands worldwide have turned to EffectiveUI as the strategic partner where the quality of the user experience is mission critical.