

AJAX

The Linchpin to a Better Web Application

AJAX is an acronym for asynchronous JavaScript and XML. It's a technique used for Web development based on the XMLHttpRequest object which is the equivalent of Microsoft's ActiveX. It is this object that allows data to be transferred asynchronously i.e. the ability to handle processes independently from other processes. Synchronous is the opposite of asynchronous, which means that processes are dependent upon other processes.

Synchronous data transfer requires the script tag to be parsed before the next element is parsed, then the next element to be parsed by the browser is dependent upon the script tag being parsed first. In effect it creates a bottleneck of one connection between Web page and browser. Style tags and link elements in the head section of the document create the same bottleneck effect. They have to be parsed one at a time before other page elements can download. Once they have been parsed then the elements in the body section can use concurrent connections to help speed up the download process. E.g. most servers handle between 2 to 4 concurrent connections between web page and browser.

Asynchronously loading methods differ from synchronous because the loading process are handled independently of each other and to a significant degree overcomes the traditional Web page design. The whole essence of this technique is to use minimal JavaScript initially and then push all the head related tags, including the rest of the JavaScript, CSS etc tags through an XMLHttpRequest object. This method uses multiple connections rather than the single one that most traditional Web pages use and consequently speeds things up considerably.

Fruition of Ajax Application

Open Standards:

AJAX is based on open standards supported by many browsers and platforms, meaning there's no fear of vendor lock-in. Most of the technologies that make up AJAX have been used extensively for years. Browsers are a trusted application platform for most users and enterprises now; this wasn't the case five years ago. One of the turning points for AJAX was the Mozilla 1.0 release that Firefox is based on and supported the XMLHttpRequest Object. This allowed the same asynchronous data transfer that had been possible in IE for years. That support and Firefox's rapid adoption really helped people understand that cross-browser rich Internet applications were possible.

Usability

Developers and designers now realize not only how the major role user-experience plays in market success, but how it affects the cost of ownership. The success of AJAX-based applications such as Google Maps over more traditional alternatives like MapQuest show that success can come to products that provide better user experience. AJAX is playing a leading role in making Web applications usable. It allows pages to request small bits of information from the server instead of whole pages. This incremental updating of pages eliminates the page refresh problem and slow response that have plagued Web applications since their inception. People have learned they need decent user interfaces and are

willing to invest in it. The bottom line here is that if users can get things done faster there's value in that whether the application is an internal intranet application, or a public Web service.

Cross-Browser and Cross-Platform Compatibility:

IE and Mozilla-based FireFox have the lion's share of the market and are arguably the easiest browsers on which to build AJAX Web applications, but now it's possible to build AJAX-based rich Internet applications that work on most modern Web browsers. This is an important reason why AJAX has become so popular. Although many developers were aware this was possibly years ago with Internet Explorer, it was overlooked because of the vendor lock-in factor. Thanks, Mozilla and FireFox.

Benefits of Regular Web Applications:

AJAX is the face of today's Web applications—and Web applications enjoy certain benefits over desktop-based ones. These include a lower cost of deployment, easier support, shorter development times, and no installation; these are just some of the benefits that have caused businesses and consumers to adopt Web-based applications since the late 90s. AJAX will only help Web applications get better and achieve more for end users.

Incremental Skills, Tools and Technologies Upgrade:

Since AJAX is based on standards that have been around for several years, many developers have now been exposed to the technologies required to build AJAX applications. This means you don't need extensive learning curve for development teams to shift from vanilla HTML and form-based applications to rich AJAX style applications. It also means that development teams working on Web applications can incrementally upgrade their user interfaces to AJAX which doesn't require a wholesale upgrade and re-write of their Web applications. Given the large investments that have been made in deploying browser-based applications since the late 90s, it's very appealing to be able to leverage existing systems and improve the user experience.

Works with Flex and Flash:

Much of the development community is locked in a heated debate of Flash vs. AJAX. There are definitely advantages and disadvantages to both technologies in different situations, but there's also a lot of synergy and opportunity for them to work together. Many developers and vendors have realized this and have implemented some really great software using both AJAX and Flash in harmony. Macromedia is also keen to see these technologies work together.

AJAX is Server Agnostic:

As much as AJAX is browser independent, it's also perfectly compatible with any standard Web server and server-side language. PHP, ASP, ASP.Net, Perl, JSP, Cold Fusion, etc. Therefore take your pick and start building. This has helped move AJAX along because all Web developers can use and talk about a common presentation layer.