

AJAX Website Applications

[Web Smart Newsletter](#) originally published November 2005 - Updated July 2006. By Eric Holter.

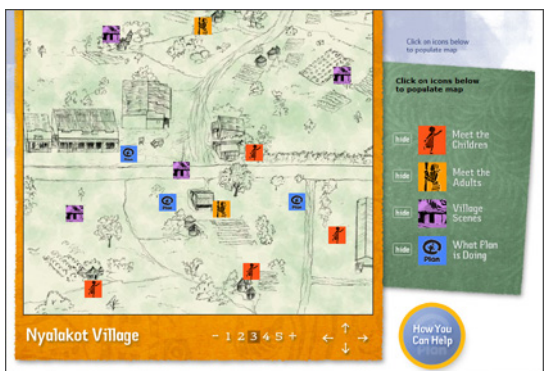
AJAX Website Applications

BREAKING NEWS: Sun Sets in West - Expected to Rise in East.

I admit it. An article about an up-and-coming web technology is hardly breaking news. In the web development biz, new technologies, languages and tools pop up every day. Most of them fade into obscurity. This year, however, a new technology has been introduced that may change how people use the web. I'll spare you the acronym for now, but this new approach to web development can make a site behave like a desktop application. This month's newsletter describes this new technology and highlights some interesting examples of its use.

When a new technology improves how the web works, we get really excited. Technology improvements rarely have discernable effects on how a website is used. In the end, what difference does it make which database or programming language is behind the scenes? Programmers get jazzed by snappier database performance, but as long a site works, normal people don't really care. But sometimes a new technology causes us to stand up and take notice.

We just completed a project using such a new technology. The technology is called AJAX. Actually, it's not really a new technology at all. Rather, it is a way of combining technologies that been around for years (mostly JavaScript, CSS, and XML if that interests you). To a web developer's joy, the fact that AJAX does not use new technology means it works dependably in all but the oldest browsers - that's music to our ears.



I'm not going to bore you with details of how AJAX works. Instead, let me show how we used it in one of our new sites. PlanUSA (www.planusa.org/maps) is an organization that provides sustainable development programs for children in poverty. Go ahead and click on the link above, then the 'Visit Now' icon so you can follow my description - you have to see it to get it.

The map that opens in a pop-up window allows you to see the village of Nyalakot with an overlay of various projects, people and scenes that you can control from a 'legend' on the left. Click any of the four square icons on the right to "show" or "hide" video links on the map for an idea of how AJAX can load new layers of information **without requiring a page refresh**. Using this tool, you can isolate on the map only the videos you are interested in. As simple as this may sound, the ability to add content to a page without refreshing is a major big deal. Let me explain.

You've probably experienced other websites where a page's information is expanded based on a click or mouse-over without the page reloading. In the past however, such effects were achieved by preloading all the information on the page during the initial page download. The content was simply hidden until a click or mouse-over triggering its display. It was there all along. The difference on PlanUSA's Village Visit is that, using AJAX, the extra layers were actually looked up from the database and added to the page on the fly, without reloading the entire page.

When select an icon from the group, the map dynamically adjusts itself to display items according to your choice, again without refreshing.

Basic AJAX Explanation

The critical contribution of AJAX

AJAX becomes critical as combinations of options grow. For example, consider the amount of data needed to select a car online. Multiply the number of vehicle types, by the number of manufactures, by the number of makes and models, by all years, by all the options like color, engine, drive, doors and so forth, and you end up with a long list of cars. Pre-loading such a list would result in an overwhelming download. With AJAX, just the first option is necessary. After the initial selection the server is called for the next set of options. The next choice calls the server again, further options are provided, and so on.

Without AJAX, the information is returned each time such database calls are made - along with all the other information needed to re-display the entire page including text, graphics, layout, and navigation. At every step along the way, each database call requires the response to return the entire modified page. That's a lot of unnecessary overhead and a lot of waiting for pages to update in between each set of choices.

With AJAX a page can ask for data from the server and get back just what it needs to update the page without reloading the whole thing.

Rise of the Next Generation of Web Applications

Desktop applications are great for activities that require real time processing and calculation. For example, as I write this newsletter in Word I get instant feedback in the form of squiggly red or green lines as I type. I don't have to refresh Word and wait for a reload every time I want to check my work. Because AJAX allows micro interactions between a page and a server without refreshing, a webpage can now function more like a desktop application.

There are already many web based applications. Online banking is a web application. Typical web applications pale in comparison to the speed and usability of desktop applications. Consider the process of updating a payee's mailing address when paying bills online. Suppose you're in the middle of entering payments when you discover a payee's address is incorrect. You click to the payee tab *refresh* select the payee *refresh* choose edit *refresh* make changes and save *refresh* click back to bill pay *refresh* and re-enter all the payments including the amount for the newly modified payee. That's at least five refreshes for a simple transaction like modifying a payee's address. When online banking implements AJAX we'll be able to edit payee information without ever leaving the bill payment page. That will be handy!

AJAX introduces significant speed and user interface improvements. A website can now function very much like a desktop application. As a result, an entire market of web based applications is now on the rise. These next generation web applications might not replace complex desktop applications like Microsoft Word with its [real time spell checking](#)--or will they?

AJAX Website Examples

Examples of AJAX sites

Now for some fun, let's take a look at some existing AJAX sites to see how much AJAX makes a difference to website interactivity and usability.

Writely (www.writely.com) - Can there be an AJAX web application to replace the sophistication of Microsoft's word processing power house? Almost! Writely is a web based word processing application. It would not be fair to compare Writely with Microsoft Word. AJAX web applications are too new to go head to head with a desktop application with the history of Word, but it really is an excellent tool. It keeps

versions of your documents as you work on them, it saves automatically every few seconds, and it has all the standard word processing features including a spell checker (although not instantaneous). Because it is web based, you can access your documents from any computer. Another advantage to a web based word processor is that a single document can be accessed by multiple people for review and collaboration. I'm not ready to abandon Word, but this is a powerful example of what an AJAX based web application can do.

Google Maps (maps.google.com) - The classic and most known example of the basic AJAX technique (although it's not pure AJAX since it uses a few different tools) is Google Maps. We've talked a lot about Google Maps in our newsletter on Local Search [Let Your Fingers Do the Clicking](#) so I won't comment on it here except to say that it is one of the best and most useful AJAX applications so far.

Google Suggest (www.google.com/webhp?complete=1&hl=en) - I also talked about Google Suggest in a previous newsletter ([What Else is Google Doing?](#)) Google Suggest is an excellent example of the power of AJAX (also not technically AJAX), but again I've described it previously so I'll move onto something new.

A9 (www.a9.com) - A9 is Amazon.com's search engine. The neat thing about A-9 is that it can search through data sources other than websites. One of those sources is, of course, book results, but it also can search blogs, people, yellow pages, movies, and much more. While doing a search if you select any of these sources, new search results feed into the page through a dynamically added column on the search results page. You can easily toggle search areas on or off without researching or reloading the page.

Instant Domain Name Search (instantdomainsearch.com) - This site is really helpful if you are trying to find a domain name for a new site. There are so few good names left, and it is extremely time consuming to enter an idea, click search, wait for the response, and try again. Instant Domain Name Search is similar to how Google suggest works. A domain's availability is dynamically determined as you type each letter. Try typing in "big..." and you'll see that big.com is, of course, taken. Continue typing "bigtoast" and you'll discover to your dismay that while bigtoa.com and bigtoas.com are indeed available, bigt.com, bigto.com, and bigtoast.com are not. Bigtoaster.com is also not available. But Bigtoasters.com is! Anyway, it sure makes finding variations and checking on the availability of potential domain names much easier.

Netflix top 100 (www.netflix.com/Top100) - Netflix top 100 is a very simple example of AJAX but it shows how even a simple idea like displaying a movie's details on mouse-over, which would be very impractical without AJAX, improves the usefulness of the page. For any given movie, displaying its description with a small image would not require much data to be pre-loaded onto the page, but x100 graphics text add up to a really long page download. AJAX makes this page more useful and interesting by pulling the data for each movie from the server and "thin streaming" it onto the page without pre-loading or re-loading.

Anticipating the Future of Web Applications

At Newfangled we are looking forward to including AJAX into a lot more of our sites. PlanUSA is one of the first, but we have more in development, and we plan on using it in our content management system and in its various mini-web applications (components) to make them faster and more robust tools for our clients. Being in the web development business is fun because there is always something new. It's even better when new technologies prove to make significant improvements in how the web works.