

Optimizing Web Applications with AJAX and Spread

a solution for ASP.NET developers

Executive Summary

Many developers are being asked to create ever more data intensive Web front ends and applications. As developers begin to realize the diverse ways in which Web technologies can be used for communicating and collaborating, technologies such as AJAX will become more prevalent. AJAX is really very simple and powerful. Integrating AJAX into your next application may solve many problems and give your users the interactivity and speed they are beginning to expect when looking for information on the Web. This is a brief overview of AJAX and how you can use it, in a spreadsheet application that has Spread, to expand the possibilities and show your customers that you have the power to solve some of their problems with ease. The use of AJAX in a Web Forms application provides many benefits and improvements. There are also limitations to its use and there are applications in which it makes more sense than others. FarPoint recommends understanding AJAX before using it in spreadsheet application development with Spread for Web Forms.

Table of Contents

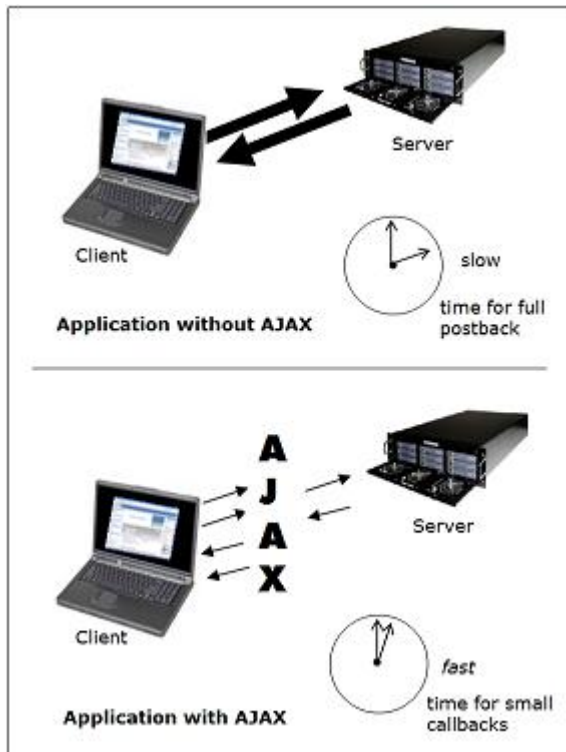
Introduction to AJAX.....	2
How AJAX Improves Web Forms.....	3
Using AJAX and Spread for Web Forms.....	4
Keeping Expectations Realistic	7
Finding More Resources	8
About FarPoint Technologies	8

Introduction to AJAX

Web technology is different from Windows-based applications in terms of speed and interactivity. While we expect short response times and numerous interactive controls from Windows applications, it is not always straightforward to implement these same techniques in Web applications. Limited by simple browsers on the client and the speed inherent in client-server communication, Web applications are mostly slower and simpler than their counterparts in the proprietary world. But developers are an ingenious bunch, and they have figured out how to use readily available technology on the Web to provide some of that same responsiveness. That available technology includes JavaScript and XML used in concert to provide asynchronous communication between the client and the server. So AJAX (Asynchronous JavaScript and XML) provides a fairly simple solution that can speed up client-server communication to allow Web pages to feel more like other applications on the desktop.

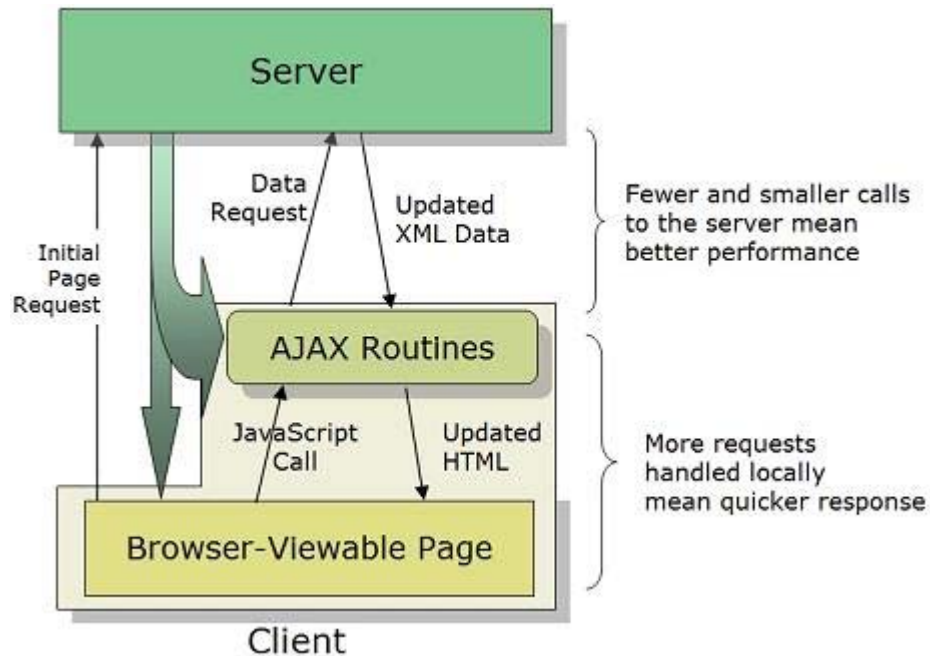
To see just how responsive Web pages can be using AJAX, you need look no further than Google Maps, <http://maps.google.com> and other sites that offer lots of data that can be manipulated with ease through an ordinary Web browser, such as having your text automatically completing at <http://demo.script.aculo.us/ajax/autocomplete> and at presenting search results as you type each character at <http://www.1976design.com/blog/>.

AJAX combines a group of technologies to provide smarter client-side pages; it is a type of programming architecture for creating interactive Web applications that can update data on a client page by making direct calls to a server without needing to go to the server for the entire page using HTML, CSS, and JavaScript. By using JavaScript routines as a middleman, AJAX sends only essential data to the server and intelligently replaces data on the client without requiring an entire page of data to be retransmitted back and forth to the server. This figure illustrates the savings in time for a postback.



How AJAX Improves Web Forms

The use of AJAX is a simple mechanism with powerful results. The experience to the end user is dramatic because it improves response time by providing a faster postback. With AJAX, when a browser makes an initial request of a page, the server provides not only the viewable part of the page but also the AJAX routines, really JavaScript functions, that will help with further page processing. Then when additional user interactions are performed, the client can perform some of that processing, handling requests or changes in data with calls to the JavaScript functions. These might handle the changes locally or send on any requests to the server for further processing. The server then can send back data in the form of XML for the AJAX routines to process. This eliminates the need in many cases for the entire page to be posted back. With large amounts of data, as in spreadsheet applications, these time savings can be impressive. The figure below shows diagrammatically the sequence of calls and how AJAX on the client side can handle much of the work.



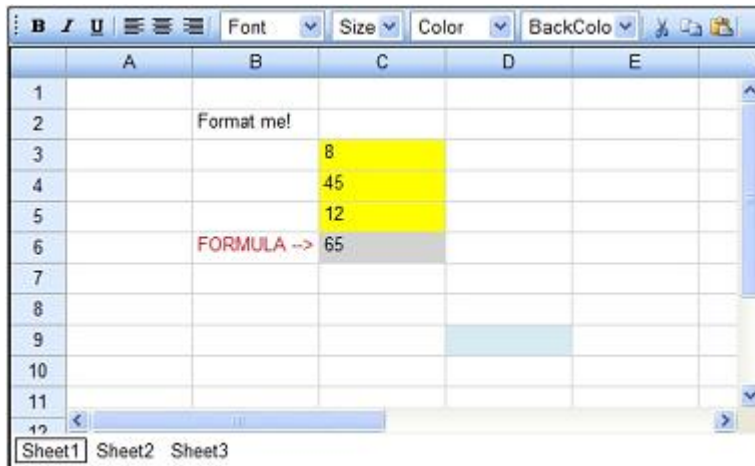
By making communication with the server less frequent and with smaller amounts of data conveyed, this asynchronous communication with the server means that processing can happen in the background from what the user sees and can happen only when needed.

Of course, this requires more work to be done up front to anticipate the type of interactions that the user may require and how to handle those interactions. But by putting some of the work on the client side, the user can have a better experience of the Web page.

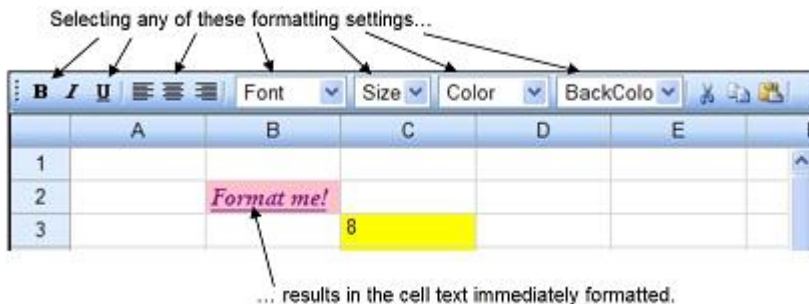
Using AJAX and Spread for Web Forms

Spreadsheets can be big, and spreadsheet applications can be some of the most data intensive applications with which to work. They can contain a lot of data and require many calculations. With components such as Spread for Web Forms, spreadsheet applications can bind to databases with millions of cells of data that must be made available to end users. For a web application, traditional ways of serving up pages can impose severe restrictions on a spreadsheet application's performance when it is bound to a large data set. Paging or moving around the sheet by the end user can demand system resource typically easily available to Windows applications but limiting Web applications. But with AJAX, those restrictions can be largely hidden from the end user as AJAX can send back to the server only those cells that have changed and updating only those calculations that need recalculating. For paging, only those rows that are displayed need to be fetched from the server.

FarPoint introduced support for AJAX in Spread for Web Forms in version 2.5. With simple Boolean properties on the FpSpread component class, Spread for Web Forms can be made aware that the application is using AJAX. See the product tour at our web site or available with the product download: <http://www.fpoint.com/netproducts/spreadweb/tour/AJAX.aspx>. The example from that product tour, with a custom toolbar to allow user interactivity, is briefly described here to show how AJAX can improve a spreadsheet application.



Here the user can format the text in an ordinary text cell and see the results immediately. Also the sum of the numbered cells can be calculated when any of the individual numbers change without having to get the entire sheet .



NOTE: The toolbar used in this demo is not a default part of Spread. It illustrates how you can use AJAX and is included in the sample with the product download but is not built in.

The code for this, on the client side, consists of functions in JavaScript for each of the formatting options in the formatting toolbar. A sample of that JavaScript code is given here.

```
<script language="javascript">

function btnOver(theTD,ftbName,imageOver,imageDown) {
    event.srcElement.style.backgroundColor = "lightsteelblue";
}
function btnOut(theTD,ftbName,imageOver,imageDown) {
    event.srcElement.style.backgroundColor = "";
}

function setFocus(ss) {
    if (document.all!=null) {
        ss.focus();
    } else {
        the_fpSpread.SetPageActiveSpread(ss);
        the_fpSpread.Focus(ss);
    }
}

function FontBold() {
    var ss = document.getElementById("FpSpread1");
    ss.CallBack("FontBold");
    setFocus(ss);
}
function FontItalic() {
    var ss = document.getElementById("FpSpread1");
    ss.CallBack("FontItalic");
    setFocus(ss);
}
function SetFontName(name) {
    if (document.all!=null) document.body.focus();
    var ss = document.getElementById("FpSpread1");
    ss.CallBack("FontName."+name);
    setFocus(ss);
}
}
```

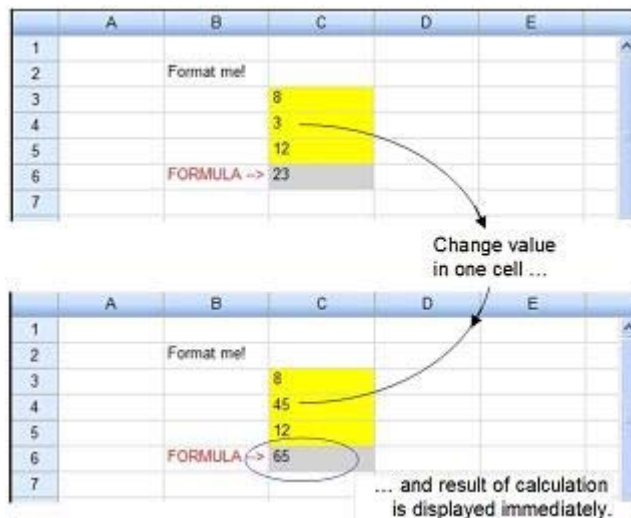
This is possible with AJAX support. Namely the EnableAjaxCall allows the web page to avoid doing a full postback and instead calling the JavaScript functions that handle the client-side updating. AJAX allows the component to refresh without refreshing the entire page. You can add AJAX support to the FpSpread component by setting the properties described here.

The **EnableAjaxCall** property of the FpSpread component affects several aspects of the operation of the spreadsheet that makes it more responsive. Built-in functions are enabled that allow for client-side operations such as expanding and collapsing child sheets in a hierarchical display, sorting columns, inserting rows, searching, and filtering. There are also ones for paging, tabbing, and other navigational aids in Spread. These are the most common uses for AJAX in Spread. By using AJAX for expanding a hierarchical display, for example, only the child sheet is needed and retrieved from the server; a full postback of the entire hierarchy is not needed. You do not have to have this property set to True to programmatically call back to Spread through AJAX, but it does enable some of the built-in functionality of Spread. Besides the flexibility to make your own callback functions, Spread provides many built-in functions for commonly performed actions that can be done with AJAX calls. A list of the functions is given on the next page. For a complete list of functions and the corresponding FpSpread component event that they raise, refer to the product documentation.

Callback Command	Client-Side Action
Add	adds a row
Cancel	Cancels an operation
CellClick	clicks a cell using the left mouse button
ChildView	displays a specified child view
Delete	deletes a row
Edit	same as Edit button
ExpandView	expands or collapses a specified row
Insert	adds a row at a particular place
Next	moves to the next page on the sheet
Page	moves to a specified page on the sheet
Prev	moves to the previous page on the sheet
Print	prints a sheet
SelectView	moves to a specified sheet
SortColumn	sorts a column
TabLeft	displays the previous sheet tabs to the left
TabRight	displays the next sheet tabs to the right
Update	saves the changes

All these are available in Spread for AJAX to use with the EnableAjaxCall property.

Another property that can be used with AJAX in Spread is the **ClientAutoCalculation** property. When this property is true, after a cell value is changed, an AJAX call is made to the FpSpread component. Then the component calculates the formulas and sends the values to the client side. The component then updates the values at the client side. This can save time if the user is only changing a value in a cell or a few cells and only a limited number of formulas need to be recalculated. With this property, the data model is updated and only those cells that are changed as a result are updated on the client. Your AJAX application can speed the operation of handling data and formulas. In the product tour, the way AJAX handles a change to a cell looks like this:



Here only individual cells are recalculated locally using the client-side JavaScript functions. Only the cells that need to be updated are sent back to the server. For large spreadsheets in which only small changes are made by the user, AJAX technology is a vast improvement.

Do not be afraid to think about all the possibilities allowed with AJAX in a spreadsheet application that has Spread for Web Forms. You can create your own functions to be used with the Callback method. This is not just a spreadsheet now, but a data management component, crunching numbers, managing data, and handling the user interface.

Keeping Expectations Realistic

FarPoint Spread for Web Forms has an object-oriented design, extensibility, and underlying models to facilitate speedy and flexible development. But AJAX will not solve all types of problems with performance issues on the client side. AJAX will not speed up everything. For example, applications with hundreds of client-side calculations may not save an appreciable amount of time with AJAX since the calculations in all these cells would still require recalculation when the data changes. With so many calculations, the client side would get bogged down and this would be better handled on the server where calculations can be performed faster with more processor bandwidth.

Also, for anything outside of the spreadsheet on the form, you would have to handle that separately. For example, if you had a message box on the form but outside of Spread that required updating when the spreadsheet changed, client-side code would have to be written to handle that and AJAX would not help in that case.

Remember also that if an AJAX callback is made, the Control OnLoad method is not called, as it would with a normal full postback. If you need it to fire, then you would need to make code that calls it and put that code in the onCallBackStart or onCallBackStopped event so that it can be run.

With AJAX, lots more thought needs to go into the development of the application. Knowing what to make interactive on the client-side and debugging the necessary AJAX code will require a bit more effort. The payoff, of course, is a better user experience and thus a happy customer. Just don't expect AJAX to do everything.

With Spread for Web Forms and AJAX, you can improve a data-intensive application. Once you have the right tools, you will be able to develop applications that may surprise even you. For more information about AJAX and Spread for Web Forms, see the resource list given on the next page.

Finding More Resources

For more detailed information about AJAX, refer to these resources:

- [AJAX: A New Approach to Web Applications \(by AdaptivePath.com\)](#)
- [Microsoft ASP.NET Developer Center: AJAX](#)
- [AJAX Tools \(Listable.com\)](#)
- [AJAX Matters](#)
- [AJAXian](#)

Conferences include [AJAX Seminar](#) and [The AJAX Experience](#).

These introductory tutorials by IBM provide some additional background information:

- [Introduction to AJAX](#)
- [Understanding Asynchronous Requests](#)
- [Advanced Requests and Responses in AJAX](#)

There are also great resources listed at the bottom of those introductory tutorial pages. And of course there's always [AJAX \(on Wikipedia\)](#) as a starting place and [AJAX Tutorials](#).

For more information on the EnableAjaxCall and the ClientAutoCalculation properties to enable AJAX support in Spread for Web Forms, see the product documentation.

About FarPoint Technologies

Whether you are developing for the Web or a stand-alone applications, FarPoint has a version of Spread that is right for you and that will deploy royalty-free in the environment you need. Founded in 1991, FarPoint Technologies, Inc. is a leading developer and publisher of professional components for Microsoft Visual Studio .NET. Our award-winning tools benefit leading corporations, software companies, and independent consultants around the world as a cost-effective solution for building distributed enterprise-wide applications for commercial or in-house use.

For more information, check out our web site or contact us directly at the North American office or European office of FarPoint:

North America Contact

FarPoint Technologies, Inc.
808 Aviation Parkway
Suite 1300
Morrisville, NC 27560 USA

Phone: 919-460-4551
email: fpsales@fpoint.com
Web: www.fpoint.com

Europe Contact

FarPoint Europe Ltd.
Whiteleaf, Roundabout Lane
West Chiltington
Pulborough, West Sussex RH20 2RL
England

Tele: +44 (0) 1798 812 372
Fax: +44 (0) 1798 813 049
email: salesEurope@fpoint.com

Information in the white paper is subject to change without notice and does not represent a commitment on the part of FarPoint Technologies, Inc.

© 2003-2006 FarPoint Technologies, Inc. All rights reserved.

Unless otherwise noted, all names of companies, products, street addresses, and persons contained herein are part of a completely fictitious scenario or scenarios and are designed solely to document the use of a FarPoint Technologies, Inc., product.

Spread for Web Forms and Spread for Windows Forms are trademarks of FarPoint Technologies, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.