Intersoft Solutions
A better web experience™

WebUI Studio.NET 2006 R2 delivers high performance AJAX-enabled components.

This whitepaper is provided as a brief reference of AJAX and its benefits, and Intersoft Solutions' initiative in delivering high performance AJAX implementation in its Web components.

Table of Contents

AJAX Explained

AJAX, shorthand for **Asynchronous JavaScript and XML**, is a web development technique for creating interactive web applications. The intent is to make web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire web page does not have to be reloaded each time the user makes a change. This is meant to increase the web page's interactivity, speed, and usability.

The AJAX technique uses a combination of:

HTML (or *XHTML*) and *CSS*, for marking up and styling information.
The *DOM* accessed with a client-side scripting language, especially *ECMAScript* implementations such as JavaScript and *JScript*, to dynamically display and interact with the information presented.
The *XMLHttpRequest* object to exchange data asynchronously with the web server. In some AJAX frameworks and in certain situations, an IFrame object is used instead of the *XMLHttpRequest* object to exchange data with the web server.
XML is sometimes used as the format for transferring data between the server and client, although any format will work, including preformatted HTML, plain text, JSON and even EBML.

AJAX is not a technology in itself, but a term that refers to the use of a group of technologies together.

In general, AJAX technique introduced several benefits as opposed to traditional web pages technique. The benefits are:

Performance

AJAX solves one of the problems with web pages - the full screen refresh when a user submits a form or clicks a link to go to another page. Depending on the design, this may result in increased performance, as smaller amounts of data can be transferred rather than whole pages of HTML, and the backend code and database calls executed are restricted to those needed to generate that data rather than generating the HTML for the whole page.

Interactivity

AJAX applications are mainly executed on the user's machine, by manipulating the current page within their browser using document object model methods. AJAX can be used for a multitude of tasks such as updating or deleting records; expanding web forms; returning simple search queries; or editing category trees—all without the requirement to fetch a full page of HTML each time a change is made. Generally only small requests need to be sent to the server, and relatively short responses are sent back. This permits the development of more interactive applications featuring more responsive user interfaces due to the use of DHTML techniques.

Intersoft Solutions' AJAX Roadmap

Introduction

Although AJAX coined its name lately in year 2005, Intersoft Solutions has been among the first ASP.NET component vendors that implement AJAX techniques back in year 2002. This AJAX technique is employed under FlyPostBack™ name which is patented by Intersoft Solutions for the implementation in initial version of *WebCombo.NET* and *WebGrid.NET Enterprise*. Until today, the FlyPostBack architecture is still widely recognized by web developers and still be used in latest Intersoft's products.

AJAX features in initial released products

Intersoft Solutions released its first product "WebCombo.NET" in year 2002 with comprehensive built-in AJAX features. The AJAX feature is enabled automatically without additional configuration and has been feature that web developers can't live without.

With the major advantages and success that "WebCombo.NET" offered to web developers and enterprise, we introduced "WebGrid.NET Professional" in mid year 2003 and "WebGrid.NET Enterprise" in late year 2003 which incorporate AJAX as the main product's feature and behavior. The implementation of earlier AJAX does not only enable "on the fly" data retrieval without full page postback in both products, it also has full support and compatibility for ASP.NET's page life cycle which was one unique innovation that Intersoft Solutions brought at that time.

The AJAX technique enables WebCombo.NET and WebGrid.NET Enterprise to deliver innovative capabilities not possible before, such as:

- Load control with large dataset in fraction of second.
- Asynchronously retrieving more data from server as user demands it.
- Submitting user inputs for server-side processing without full postback.
- Sending a custom function-call request from client-side and returning information from server-side without page refresh.
- More than 75 data-related features are functioning without full postback, from simple grouping, sorting, filtering, hierarchical child population, self referencing child population, virtual load with customizable load on demand option, and more.

AJAX as an integral part of features in WebUI.NET Framework

Intersoft Solutions has been committed to deliver extremely reusable, high performance and rich components to help developers building a better user experience of web application. In early year 2004, we take the initiative to bring a new concept of component architecture that solve major limitations in reusability. The new architecture named "WebUI.NET Framework" takes the benefits of framework's methodology which enables top-level products to access common shared codes and providing generic user interface as well as object model consistency.

"WebUI.NET Framework was architected with comprehensive set of features including rock-solid AJAX interface as one of the design goals. Since the availability of WebUI.NET Framework, all products have been redesigned to build upon WebUI.NET Framework to take advantage of better AJAX implementation."

The reinvention of AJAX in WebUI.NET Framework introduces several benefits and new capabilities related to AJAX:

AJAX-enabled products such as WebCombo.NET 3.0 and WebGrid.NET Enterprise 4.0 can use the same interface and single source of codes for every AJAX operations. This exposes better interoperability and deeper integration between products as well as behavior's consistency.

Cross-browser AJAX support through IEMozBridge™ architecture. Supported browsers include all Mozilla-derived browsers such as Firefox, Netscape, Camina, etc.

Innovative ClientAction™ engine which acts as the bridge between server and client. The ClientAction engine allows you to call client side actions from server side code during AJAX events. For instance, invoking a client-side function, showing messagebox or navigating to another page.

Automatic view state persistence enables the state of controls which have been modified during AJAX events to persist in the subsequent postback request.

Per-control AJAX settings. For even greater control, each component's instances can have different settings over what data to be collected and transmitted during AJAX operation. This enables developers to customize the *FlyPostBackSettings* configuration for achieving highest performance AJAX calls.

Supports for emerging AJAX Framework such as Microsoft "Atlas".

Supports for latest Microsoft ASP.NET 2.0.

Cutting-edge UI components on the top of AJAX

As the leading vendor in web user interfaces and specifically in the expertise of AJAX since year 2002, Intersoft Solutions introduced a set of cutting-edge UI components in late year 2005. These next generation UI components are branded under one product family "WebDesktop.NET". Built upon WebUI.NET Framework, WebDesktop.NET automatically takes advantage of all AJAX features available in WebUI.NET Framework.

One of most remarkably cutting-edge UI components with built-in AJAX capabilities available in the WebDesktop.NET 1.5 is *WebNotification* component. The notification component sports the look and feel of desktop's notification similar to Microsoft Outlook® or MSN Messenger® and powered with AJAX technique for asynchronously retrieving new notifications from server-side periodically in configured time interval.

The state-of-the-art UI components together with AJAX in Intersoft WebUI Studio.NET Suite enables you to:

Develop **R**ich **I**nternet **A**pplications rapidly using professional approach.
Deliver responsive and intuitive desktop-style web applications.
Boost user experience with smooth and elegant web interfaces.
Get ready toward "Web 2.0".

Today's AJAX for universal client/server communication

Intersoft WebUI Studio.NET 2006 R2 includes a new standalone AJAX Manager designed to address several limitations exposed by built-in AJAX implementation. This new AJAX Manager named "WebFlyPostBackManager.NET" is built around the FlyPostBack™ architecture, and is a member of *WebDesktop.NET 1.5* product family.

Unlike most AJAX wrappers in the market, WebFlyPostBackManager is a powerful and rich AJAX Manager that can be used in 2 modes:

Data Exchange
Use the component to retrieve server-side resources from:
1) *XML Web Services*
2) *ASP.NET Web Form*

You can easily invoke the server-side methods in the client side by using its method name directly. For instance, you have a method named "GetCustomerData" in a WebService page. You can now invoke the method and get the result in the client side by using
ajaxManager.GetCustomerData();

The component accepts basic and advanced data types that returned from the server side. This enables developers at all level to leverage innovative, new type of web application that is more responsive and intuitive.

The supported data types are:

- *Basic types*, such as string, integer, float, datetime, Boolean, and array.
- *Advanced types*, such as custom objects, complex objects with nested objects.
- *Collection types*, such as collection of custom objects.
- *Dataset types*, such as DataSet, DataTable or DataView objects.

User Interface

Use the component to refresh standard server-side controls using AJAX technique. The User Interface mode can be operated in the following two methods:

1) *Manual Programmatic method*.
   Each WebFlyPostBackManager control has its associated client-side object, which can be used to initiate an AJAX callback request via client-side method. You can supply either simple or complex object as the parameters as necessary. The call invokes the specified server-side method which contains codes that modify the desired server-side controls. Thanks to ClientAction™ engine, you can use strongly-typed object model for modifying server-side controls or performing client action without has to deal with low-level *HtmlTextWriter* API. The modified controls will be send back to client, and partially updated without reloading the entire page.
2) *Automatic method*.
   While using manual method is more suitable for advanced developers who want to gain more controls over the AJAX call, working with numerous AJAX calls in the same page can be cumbersome. WebFlyPostBackManager comes with innovative feature designed to enable automatic user interface refresh by simply turning one property *EnableUIMode* to true.

The recommended method for User Interface usage scenario is *Automatic method*. The Automatic method introduces several advantages and unique capabilities such as:

- No JavaScript codes is required. Normally, manual AJAX call from client-side codes is required in order to invoke certain server-side method. With this Automatic feature, the control will manage the necessary AJAX calls automatically.
- Writes server-side codes normally as in traditional full postback mode. That means you don't need separate methods to serve the AJAX calls. You simply put your codes inside the event of the server-side control. For instance, you can put *textBox1.Text = "New content from AJAX Call"* inside *Button1_Click* server side event.
- Works with any standard ASP.NET controls. You can update the content of any standard ASP.NET controls such as adding new option to DropDownList, showing new data in GridView, or simply change the text of a Label – all without full reload.
- Automatic "delta" rendering. The AJAX Manager is able to detect only modified controls during the AJAX call, and update the modified controls partially.
- Does not need a container or wrapper. The controls that you want to update in the AJAX call can reside anywhere in the page without the need to be placed in a special container or template which exposes usability and performance issues.
- Selective controls update. The AJAX Manager provides a configuration to let you specify which controls should be checked for update during the AJAX call.
- Viewstate persistence. The latest page's viewstate made by the AJAX call will be automatically persisted, so that the subsequent postback can reflect to the latest state made during AJAX call.

Uniqueness of Intersoft's AJAX implementation

With its expertise around Web technologies and AJAX in particular, Intersoft Solutions have implemented the AJAX mechanism to all its products since year 2002, three years earlier before AJAX term coined.

Since the first AJAX implementation in our initial products, we have invented unique AJAX architecture which consists of:

Strong interfaces and OOP-style in both server-side and client-side implementation.
This enables all top-level products to use same interface for every AJAX operations which brings ease-of-use and reduced learning curve for developers while working with our products.
Automatic control state persistence.
Every user elements and objects that are modified during user interactions or through programmatic client side API are automatically persisted to server-side during AJAX or standard full page postback. This innovative feature was the first of its kind in the industry, which implemented in WebCombo.NET and WebGrid.NET Enterprise back in year 2003.
Professional implementation.
Our professionally implemented AJAX architecture increases developer's productivity by reducing the lines of codes required for the AJAX call. Furthermore, developer can use elegant, strongly-typed object model of server-side codes accompanied with rich client-side API for building sophisticated web application that highly maintainable and extensible in the future.
Advanced support for Microsoft ASP.NET® and emerging technologies such as "Atlas".
We have fully supported Microsoft ASP.NET's page life cycle and solved numerous AJAX-specific limitations and problems since our initial version of AJAX implementation.

These approaches altogether introduce unique benefits that other AJAX Framework may not have, such as:

High level of reusability.
Deeper interoperability.
Consistent behaviors and same-set of features.
Small footprint of codes since all products are using single-source AJAX architecture.
Reduced memory usage and avoided leaks.

The uniqueness of our AJAX implementation also heavily relies on our deep understanding of the AJAX concept and the most efficient usage of it. Many AJAX Frameworks have failed to deliver responsive user interface, but even worse making the application slower. That failure is possible because the vendors makes *excessive use* of AJAX calls in inappropriate places and unsuitable scenarios. For instance, one of the worse AJAX implementation sample that we have ever seen is by sending "mouse move" events as AJAX call which makes the application terribly slow because the mouse move events generated approximately 50-70 AJAX calls at once.

Although AJAX is a great technique and a foundation for creating smoother user experience in web pages, AJAX does not mean to be used brutally which lead to incorrect designs.

Most AJAX-enabled components simply render back the whole control's HTML output during AJAX call. For instance, switching from edit mode to normal view in a DataGrid in AJAX callback will send back the whole HTML output for the DataGrid. This is not an efficient approach and does not expose the true advantage of using AJAX technique, because the whole HTML output of the DataGrid is send back to client-side although only "small pieces" of data that may have been changed.

With regards of this topic, we would love to share one of unique AJAX implementation in our WebGrid.NET Enterprise product. Our WebGrid component offers approximately over 75 data-intensive features which built upon AJAX technique. One of the most important tasks is updating, adding or deleting certain rows. While these data modification take place, our WebGrid sends only information about that specific rows to the server-side for further processing. After the server-side processing, WebGrid returns only updated rows to the client-side instead of the whole control's HTML output.

"Our AJAX implementation goes beyond simply providing *simple AJAX callbacks*, that is oftentimes too general (simply providing low-level abstraction for asynchronous calls). We take pride in delivering *higher-level abstraction* of functionality using the AJAX technique, and specially focusing on the design methodology and performance to bring greater business value for our customers."

AJAX implementation reference in WebUI Studio.NET 2006 R2

Please visit our Live Demo at http://www.intersoftpt.com/WebDesktop/Live to experience the samples online. You can also run the sample from your local computer at http://localhost/WD15Samples if you already installed WebUI Studio.NET 2006 R2.

Following is a list of several examples that used AJAX technique in the aforementioned sample:

Write Email is a sophisticated mail composing user interface implemented using WebMenuBar, WebToolBar, WebButton, WebInput and WebCombo. Typing into "To" field populates data from the server-side against matched input in AJAX-style loading. The "To" input is using WebCombo.NET component that implements built-in AJAX capabilities.
Change Desktop Style is an example designed to change the entire web application's look and feel globally. The "Apply" button is using AJAX callback which reads the selected theme in the server-side and apply the theme accordingly.
My Application is showcasing a new model of web application's user interface that simulate desktop application in the term of user interactions, layout and visual appearances. After the first load, a WebGrid should appear in the right-side pane. You can try several user interactions to the WebGrid, such as resizing a column then sorting, grouping multiple columns, reordering the grouped columns using drag and drop or ungroup them. Notice that all actions in the WebGrid are done through AJAX technique, also every objects that changed during user interaction are automatically persisted.

Intersoft Solutions
A better web experience™

Future of AJAX – the Intersoft's way.

At Intersoft Solutions, we are strived to continuously research and deliver unique innovations around Web user interfaces and AJAX in particular to bring the best value and quality of products for our customers.

With our current offering and standing in the marketplace, we have go beyond several milestones ahead of other AJAX vendors. Furthermore, we are planning to include new capabilities around the AJAX implementation in future products, such as:

- AJAX-powered client side data components.
- Implementation of AJAX in visual data representation and OLAPS component.
- Integration and interoperability with future Web technologies.
- Real-world AJAX examples combined with cutting-edge UI controls that are more advanced and sophisticated in functionalities and user experience. For instance, live customer service implemented through *WebDesktopManager* and *WebFlyPostBackManager*.

Should you have questions or feedbacks regarding our future AJAX approach or about our products mentioned in this white paper, please feel free to contact us at feedback@intersoftpt.com

Intersoft Solutions Team

Delivering A Better Web Experience™