BACKBASE

Rich Internet Application Strategy

"AJAX and Beyond"

Backbase B.V.
Stephensonstraat 19
1097 BA Amsterdam
the Netherlands
+31 (0)20 750 7305
info@backbase.com

Rich Internet Applications: "AJAX and Beyond"

Although Dynamic HTML technology has been around for some years, Google's new web applications, Suggest and Map, have certainly boosted the awareness of a model based upon DHTML / W3C standards for creating Rich Internet Applications.

Adaptive Path's Jesse James Garret gave this model a name: **AJAX**. Compared with classic page-based web applications, AJAX introduces a new web presentation tier model that is different in three important ways, namely:

1. Use of a client-side engine as an intermediate between the User Interface (UI) and the server;
2. User activity leads to JavaScript calls to the client-side engine instead of a page request to the server;
3. XML data transfer between server and engine.

Alternately stated, AJAX solutions contain a client-side engine, consisting of JavaScript functions, which renders the user interface and communicates with the server in XML format. This engine sits in the web browser and does not require a plug-in or user-install.

The formalization of this model has made it clear to developers that it is possible to develop Rich Internet Applications based on standards and thus sparked an interest in doing so.

There are numerous examples on the Internet today of custom widgets (controls) for improving interaction in the UI. These are usually tucked away in frames with complex JavaScript coding techniques which are very complex to manage and introduce extensive redundancy as 90% of the code is comparable but not necessarily the same. In an interview posted on Infoworld.com, John Udell states:

"We clearly need more advanced widgetry to help us deal with a range of data types and to guide us through sophisticated interaction scenarios".

In order to deliver a manageable and scalable RIA solution, we need to go beyond the custom widget development of the current AJAX model. In 2002, Backbase began to develop a new **generic client-side GUI management engine** (the Backbase Presentation Client) combined with a **declarative tag-based UI language**: BXML ("Backbase eXtensible Markup Language").

The Backbase Presentation Client (BPC) is based completely on AJAX technologies, but differs via inclusion of a generic User Interface (UI) declaration language (BXML).

Both the Backbase Presentation Client and BXML are designed to kick-start Rich Internet Applications projects using normal HTML technologies.

For sample applications visit: http://www.backbase.com.

Following, the Backbase Presentation Client and BXML are described in more detail.

Backbase Presentation Client (BPC)

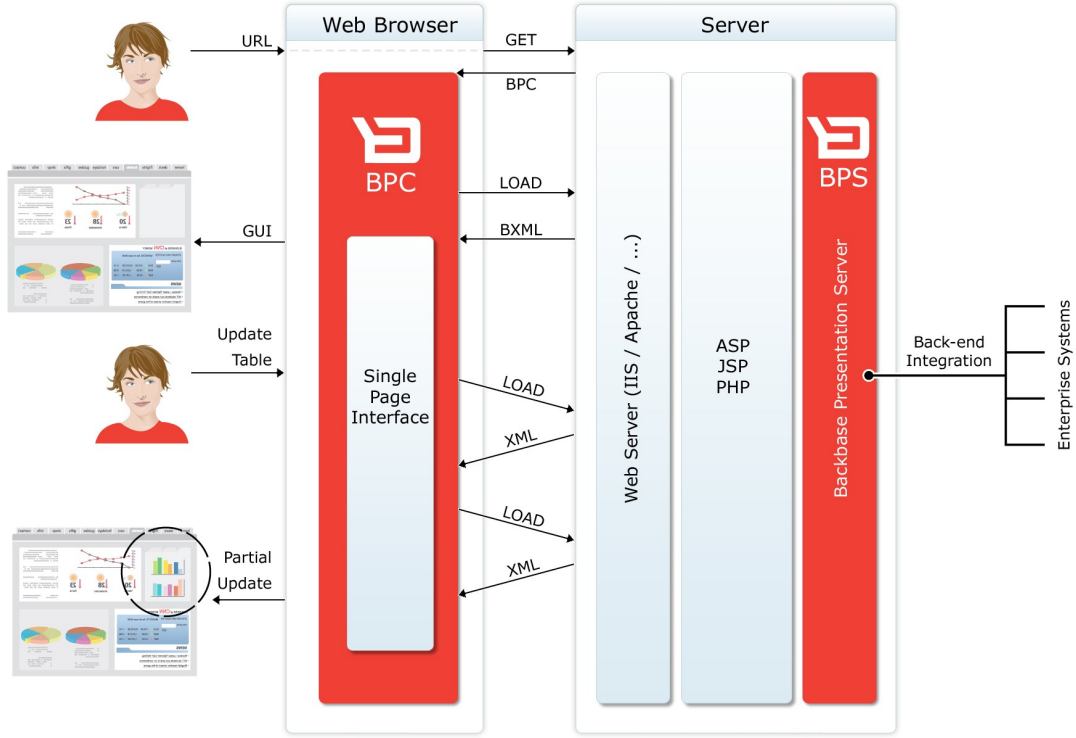
The BPC is a robust and generic client-side AJAX engine that renders Rich Internet Applications at runtime.

When a Backbase-enabled application/site is requested, the browser loads the ultra-light (30 KB. compressed) BPC (script type="text/javascript" src="../bpc/boot.js"). This zero-install process is seamless to the end-user (i.e. there are no plug-ins or applet downloads required).

BPC interprets the BXML tags and transforms them into the corresponding DOM structure, keeping full control over aspects such as: layout, state, asynchronous communication, data binding, events, relationships, hierarchy and many more.

Figure 1 below gives an overview of the components in relation to an interaction process.

Figure 1 – Backbase solution overview



Once activated within the web browser, the BPC creates a client-side presentation framework, which includes facilities for state management, event management, client-server synchronization and incremental data loading. This framework utilizes client-side resources to directly execute the web application's interaction and presentation logic and allows for the instantiation of Single Page Interfaces.

Single Page Interfaces eliminate the feel of 'page flipping' as experienced with the traditional HTML model. End-user interactions results in data requests to the server instead of page requests, see Figure 1 above. This results in partial updates of the User

Interface. The single page offers end-users immediate response, smooth transitions between states and continuous, stable workflows.

Functions of the Backbase Presentation Client:

- Create and manage a Single Page Interface;
- Declare Rich User Interfaces with BXML, enriching XHTML code;
- Utilize XHTML and CSS standards for presentation;
- Utilize JavaScript to extend BXML declarations with custom logic;
- Dynamic data interchange and manipulation using XML and XSLT;
- Manage client-side events (user interaction);
- Manage client-side GUI states;
- Manage overall GUI composition and relationships (data binding);
- Manage GUI layout (integrating re-usable views / client controls);
- Manage GUI behavior (combine events, views and data-bindings);
- Manage overall client-server communication (Asynchronous Parallel Processes);
- Utilize client-side integrations with Web Services (SOAP, BPEL, XML-RPC);
- Deliver a cross-browser & cross-platform compatible development environment;
- Deliver out-of-the-box client controls (e.g. widgets to kick-start RIA projects).

In addition to the Backbase Presentation Client, there is also a BackBase Presentation Server (both for JAVA and .NET), a Backbase Development Environment (both JAVA and .NET) and adapters for specific Application Servers available.

Backbase eXtensible Markup Language (BXML)

Developing and testing (X)HTML- and JavaScript based web applications can be extremely labor intense. While web browsers have progressed in terms of their support for interactivity through the Document Object Model (DOM), JavaScript, and XHTML, they still lack a formal UI language. The lack of a formal UI language has forced developers to perform a lot of custom and labor intensive UI development efforts (especially with JavaScript).

In order to solve this, Backbase provides a formalized UI declaration language (the Backbase eXtensible Markup Language – BXML). BXML is an XML-based markup language which provides tags (in namespaces) that enable developers to declare highly interactive Rich Internet Applications and manage the data-exchange between the server and client:

- xmlns:b=http://www.backbase.com/b for visual tags.
- xmlns:s=http://www.backbase.com/s for systems tags.

BXML is fully based on industry standards (W3C) such as XML, XHTML, CSS and DOM, and therefore works seamlessly with common web browsers. In addition, BXML offers developers a cross-browser cross-platform compatible environment, creating consistent user experiences.

BXML offers advanced capabilities for user interface declaration, such as controls (e.g. buttons, toggles, windows, dialogs, menus, lists, tables, decks, tabs, sliders, intelligent forms) and behaviors (e.g. move-to, open, close, collapse, display, hide, drag, receive). This functionality, traditionally developed via cumbersome, hard-to-maintain JavaScript code, is easy to implement with BXML. Via the BXML tags web developers can easily manage objects that represent the rich user interface. All rendering and parsing of the User Interface is executed directly in the web browser (utilizing the Backbase Presentation Client).

BXML extends (X)HTML, anyone with (X)HTML skills can quickly and easily augment existing (X)HTML code with BXML code (both of which are declarative and tag-based). This allows developers to evolve their current Internet Applications piece-by-piece into Rich Internet Applications via the inclusion of more expressive BXML tags.

With BXML web application developers have an easy to learn, yet very powerful language that helps them create advanced Rich Internet Applications. BXML enables web developers to protect their current investments (e.g. in DHTML technology and knowledge) and allow for a step-by-step transition from traditional HTML-based interfaces in to fully functional Rich Internet Applications.

Furthermore, BXML is extensible, meaning developers can create their own custom client controls (i.e. attributes can be defined for lay-out, content and behavior).

BXML Code Samples

1) Navigation box:

This sample shows the use of a control called "**navbox**", useful for showing and hiding levels of sub-navigation and further detail.

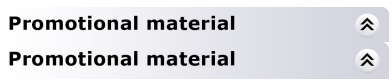
The code sample also shows the basics of a BXML page:

- Valid XHTML ;
- Namespace definitions for the BXML vocabulary (b: and s:) ;
- Loading the BPC JavaScript library ;
- Use of B-Tags ;
- Combination of B-Tags with standard XHTML tags.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                                    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xml:lang="en" xmlns=http://www.w3.org/1999/xhtml
        xmlns:b=http://www.backbase.com/b  xmlns:s=http://www.backbase.com/s>

        <head>
                <meta http-equiv="Content-Type" content="text/html; utf-8" />
                <title>NavBox</title>
                <script type="text/javascript" src="../bpc/boot.js"></script>
        </head>
        <body onload="bpc.boot('../bpc/');">
                <xmp>
                        <b:navbox style="width: 20em;">
                        <b:navhead>Promotional material</b:navhead>
                                <b:navbody>
                                 Our collection of promotional material for this movie consists of:
                                    <ul><li>Taglines</li>
                                        <li>Trailers</li>
                                        <li>Posters</li>
                                        <li>Photo gallery</li> </ul>
                                </b:navbody>
                        </b:navbox>
                        <b:navbox style="width: 20em;">
                        <b:navhead>Promotional  material</b:navhead>
                        </b:navbox>
                </xmp>
        </body>
</html>
```

Figure 2 – BXML/XHTML navigation box sample.



Initial situation.

After selecting the arrow of the navbox.

2) Drag & Drop:

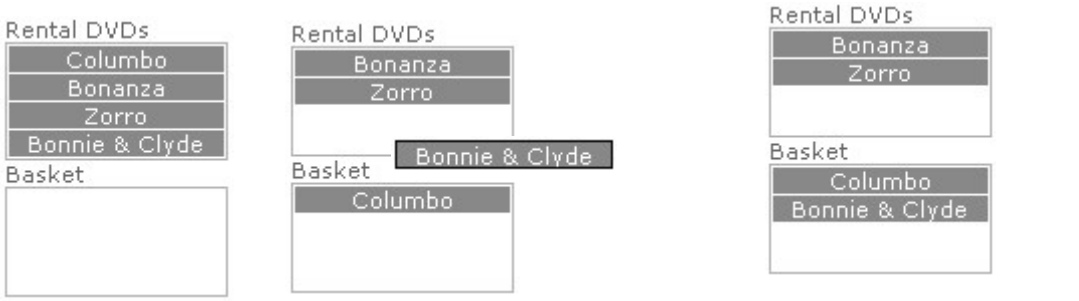
The following example shows the use of BXML tags to enable drag & drop functionality:

- Attribute "b:drag" to enable dragging ;
- Attribute "b:dragreceive" to allow dropping.

Drag & drop in this way is enabled with standard XHTML elements in a declarative way.

```
<div>Rental DVDs</div>
<div b:dragreceive="DVDrentals" class="dragreceive">
<div b:drag="DVDrentals" class="dragitem">Columbo</div>
<div b:drag="DVDrentals" class="dragitem">Bonanza</div>
<div b:drag="DVDrentals" class="dragitem">Zorro</div>
<div b:drag="DVDrentals" class="dragitem">Bonny & Clyde</div>
</div>
<div>Basket</div>
<div b:dragreceive="DVDrentals" class="dragreceive"></div>
```

Figure3 – BXML/XHTML drag & drop sample



Initial situation. Drag/Drop items into the basket. Basket filled with items.

Backbase offers an extensive library of Rich UI functions (BXML). Given the declarative nature and the extensibility of the language it is easy to write (just extending existing HTML code) and easy for developer to create their own Custom Client Controls.