*Ajax Panel*

Ajax has become a mandatory feature in each and every web application. Rarely to find a website does not use at least one Ajax call. Despite the fact the magic idea hit the cyber in 2005; it is something we cannot imagine the virtual world without.

## 1. Overview

JSPX is supporting creating Ajax web forms as easy as it never been. As basic principle in JSPX, providing new features should not affect any previous features. Also it should be easy to use.

Ajax panel is a jspx control that is presented to provide Ajax support to jspx forms. Ajax panel is used by wrapping set of controls with the tag <ajaxPanel>. Only this will do the trick and the form will be converted to submit in Ajax manner, and this is applicable on both normal and file upload.

## 2. Ajax or Not

The most powerful feature jspx is giving while providing Ajax; is the flexibility of adding or removing the Ajax from a normal webform. It does not require adding neither any extra libraries nor any change on the project. It is simple as wrapping the section of HTML that is going to be Ajax with the Tag AjaxPanel.

An example of this,

```
<form id="myForm" method="post" >
<input type="password" id="somePassword"/>

<ajaxPanel id="ajaxPanel1">
      <table>
            <tr>
                  <td >
                        <input type="text" maxlength="19" size="39"
id="serial"/>
                  </td>
                  <td >
                        <input type="button" id="button" group="x"
value="Save Serial" />
                  </td>
            </tr>
      </table>
</ajaxPanel>

</form>
```

In this example, when clicking on the save serial button, only the section within the ajaxPanel tag will be updated with the response coming from the server. When running this example, you will find that the password field will keep its value. That is indicating that it was not refreshed.

On the server side, you will not need to change anything from the usual code you built before. Neither when adding ajaxPanel nor when removing it.

For more examples please download the demo project from this link:
http://jspx-bay.sourceforge.net/index.html?l=pages/tout/demo.html

## 3. Postback and render

When using ajsxPanel the postback operation is changed and altered by using the XMLHTTP object in your browser.

The postback data sent to the server is the same data collected when the browser is submitting non-Ajax call. This provides the ability to your code to interact with whatever data in other controls that are filled with user and don't lie within the ajaxPanel.

This makes it seamless to interact with the request from your server side code as whether it is an Ajax or Not.

On the other hand, while rendering the response, only the ajaxPanel content will be rendered. This improves the performance of the operation and reduces the time consumed.

## 4. Multipart/Form

One of the biggest challenges that Ajax is facing is the Mulitpart/Form which is the ability to use Ajax to upload files to the server. Looking into existing mail porviders like Gmail, Windows Live mail and Yahoo mail. We see that uploading attachments is easy and looks the same as using Ajax.

The fact behind this is that Ajax is not using at all while doing so. Instead a mixed technique is used including iFrames and JavaScript.

Jspx provides the ability to apply ajaxPanel on both normal forms and mulitpart forms without a single change in neither your code nor your HTML. Seamlessly, JSPX provides the required script to submit both normal form and multipart from in Ajax manner.

So, the above example is working on both cases.

## 5. Auto refresh

Dynamic websites always have a very frequently changing content. Some features like notification services in a website or a news ticker needs to be refreshed on a periodic manner. Such feature requires a lot of JavaScript and server side code.

This feature is supported in jspx in a very simple manner. Ajax panel can be set to automatically post its content at certain constant time. This provides a better user interactive way to refresh the content of

your page periodically. Using this feature is very easy, just by setting the attribute refreshtime to number of milliseconds.

You will need also to provide event handler for the refresh event. This is done by setting the attribute onrefresh to the name of the method stub that is handling the event.

```
public void refreshPanel(WebControl invoker, String args)
```

So every time the ajaxPanel is refreshed, the server side event handler refreshPanel will be invoked, where you can alter the content of the panel and displaying the fresh items.

```
<ajaxPanel id="ajaxPanel1"  onRefresh="refreshPanel"  refreshtime="3000">
      <table>
            <tr>
                  <td style="color: blue; font-style: italic;">
                        this text will be updated each 3 seconds.
                  </td>
                  <td >
                        <label id="result"></label>
                  </td>
            </tr>
      </table>
</ajaxPanel>
```

## 6. Ajax Loading

As the Ajax action will happen without notifying the user, the user may not feel that his action has been fired. Hence displaying loading sign will give the feeling of an action under processing. You can provide loading template with any html you wish to display. This is done by using ajaxloading control. Within this control you can provide your own loading sign.

```
<ajaxPanel id="ajaxPanel1"  onRefresh="refreshPanel"  refreshtime="3000">
      <ajaxLoading>
            <div style="background-color: red;width:70;">
                   loading....
            </div>
      </ajaxLoading>
      <table>
            <tr>
                  <td style="color: blue; font-style: italic;">
                        This is a content that is obtained using Ajax.
                  </td>
            </tr>
      </table>
</ajaxPanel>
```