


# MAX 2006 Beyond Boundaries



Donald Booth  
Dreamweaver QE/Spry Team  
Adobe Systems, Inc.

2006 Adobe Systems Incorporated. All Rights Reserved. 1

## Building a Spry Page

1. Attach CSS/JS
  1. Browse to the Assets folder and attach 'max.css'.
  2. Attach the 2 js files.
 

```
<script src="SpryAssets/xpath.js" type="text/javascript"></script>
<script src="SpryAssets/SpryData.js" type="text/javascript"></script>
```
2. Create Dataset
 

Before we create the dataset for our page, let's discuss and visualize the data.

  1. Pull in XML data. The 'data' folder contains our XML files.
    1. Create a data set by typing:
 

```
var ds1 = new Spry.Data.XMLDataSet("data/mercury.xml", "missions/mission");
```
    2. Set Column Types. Setting the column types allows us to sort properly.
      1. Set column type for date:
        - `ds1.setColumnTypes("@flight", "number");`

2006 Adobe Systems Incorporated. All Rights Reserved. 4

## Spry Overview

- Why did we make Spry?
  - High Entry Point
  - Require Advanced Coding, therefore
    - Only available to developers
- Our Goal
  - Lower the entry barrier
  - Keep the code clear
  - Use established standards
  - Let the framework do the lifting

```

else
{
  var curDataSetRow = dsContext.getCurrentRow();
  if (curDataSetRow)
  outputStr += curDataSetRow(token.data);
}

```

Into:

```
Spry:even="class"
```

2006 Adobe Systems Incorporated. All Rights Reserved. 2

## Create Spry Page

1. Add a Spry Region
  1. Add Div to Page and add the spryregion attribute.
 

```
<div spryregion="ds1">Content</div>
```

At this point, we should test the page.  
Add (date) to the DIV. Preview in Browser. We should see the first date from the XML file. If not, check the code.
1. Insert a Table
  1. Within the DIV, insert a Table with 2 rows and 4 columns.
  2. To the second row, add data references.
 

```

$ (@flight)
$ (mission_name)
$ (ship_name)
$ (date)

```
  3. Add column names to first row.
  4. Add spryrepeat to the second TR.
 

```
<tr spryrepeat="ds1">
```
  5. PIB. We should now see the full table.

2006 Adobe Systems Incorporated. All Rights Reserved. 4

## The Spry Trinity

- Spry Data
  - Data Sets
  - Custom Attributes
  - Data References
  - Regions
  - Repeating
  - Functions
- Effects
- Widgets

2006 Adobe Systems Incorporated. All Rights Reserved. 3

## Building a Spry Page, pt. 2

1. Add Sorting to columns
  1. Add to the respective TH:
 

```
ds1.sort('flight_name','toggle');
```
  2. Repeat for other columns
 

```

$ ds1.sort('@flight','toggle');
$ ds1.sort('ship_name','toggle');
$ ds1.sort('date','toggle');

```
2. Add behaviors
  1. Again, to the repeating TR, add a spryhover:
 

```
<tr spryrepeat="ds1" spryhover="rowHover">
```
  2. Add spryselect with 'rowSelect'.
 

```
<tr spryrepeat="ds1" spryhover="rowHover" spryselect="rowSelect">
```
  3. Add data reference for alt row colors.
 

```
<tr spryrepeat="ds1" spryhover="rowHover" spryselect="rowSelect" class="ds_EvenOddRow">
```
  4. Preview in Browser. The table will have alternating row colors, change color when you are over a row and a different color when you click a row.

Everyone up to speed?

2006 Adobe Systems Incorporated. All Rights Reserved. 6

## Spry Page, pt. 3

1. Add a detail region
  1. Use the Draw Layer object to create a layer on the right of the page. Insert a sprydetailregion with dataset name.
 

```
<div id="ap1" sprydetailregion="ds1">content</div>
```
2. Add data to new region
  1. Add data references. To the detail div, add:
 

```
{flight_name}
{ship_name}
{date}
```
  2. Add image link
 

```

```
  3. PIB.

The detail region should show the first row of the data.  
Let's update the page so that clicking a table row causes the detail region to update.

1. Add data set update behavior
  1. To the repeating TR, add:
 

```
onClick="ds1.setCurrentRow('{ds_RowID}');
```

2006 Adobe Systems Incorporated. All Rights Reserved. 7

## Spry Accordion

- Building an accordion
  - Attach script
  - Container DIV
    - Panel DIV
      - Label DIV
      - Content DIV

But...the only important thing in the structure. Tags can be any block level element.

- Container DIV
  - Panel DIV
    - Label H2
    - Content P

2006 Adobe Systems Incorporated. All Rights Reserved. 10

## Create Spry Page

- Preview the page. Clicking on a row of the table should cause the detail region to update!
- We will come back to this page later.

2006 Adobe Systems Incorporated. All Rights Reserved. 8

## Spry Accordion

1. Start by attaching the SpryAccordion script. In the head of the page, add:
 


```
<script type="text/JavaScript" language="JavaScript" src="assets/SpryAccordion.js"></script>
```
2. To the body, we will construct the accordion markup.
  1. Add the main container div. This will hold all our panels.
 

```
<div id="accordion1"></div>
```
  2. Now add the panel div structure within the above div.
 

```
<div class="accPanel">
  <div class="accTab">Tab</div>
  <div class="accContent">Content</div>
</div>
```
  3. Now copy the panel div and paste the panel div below the first to create a second panel. Change the content of the tab and content divs so it is clear when the panels change.
  4. Repeat to create a third panel.

2006 Adobe Systems Incorporated. All Rights Reserved. 11

## Spry Widgets



- Let's look into the widget model.
- Clean markup.
- Structure based, not tag based.
- Small script to construct.

2006 Adobe Systems Incorporated. All Rights Reserved. 9

## Spry Accordion/Functions

1. Add script anywhere **below** the accordion markup. Use the container ID in the constructor
 

```
<script language="JavaScript" type="text/javascript">
  var sampleAccordion = new Spry.Widget.Accordion("accordion1");
</script>
```

Since we use simple markup, accordions can be styled with regular CSS.

  - Preview in Browser. The accordion should be functional.

**Accordion Functions**

Accordions have functions that allow you to control the accordion programmatically.

  1. Add 2 buttons to the page.
 

```
<button >open next panel </button>
<button >open prev panel </button>
```

2006 Adobe Systems Incorporated. All Rights Reserved. 12

## Accordion Functions



1. To these <button> tags, add the following code:

- <button onclick="accordion1.openNextPanel();" >open next panel </button>
- <button onclick="accordion1.openPreviousPanel();" >open prev panel </button>

2. Preview in Browser and test the accordion and the new buttons!

Let's play with the idea of widgets as structure.

- Try changing your accordion tab tags from <div> to <h3> tags.
- Change the content <div> to a <p> tag. PIB and it still works!



## Spry Effects



1. To the <a> tag, add the following code:

```
<a onclick="Spry.Effect.Blind('Accordion1',{duration:1000,from:'100%',to:'0%',toggle:true});">
```

2. Preview in Browser. Clicking the link should hide the accordion. Clicking again should show it. The accordion should still be functional.



## Spry Effects



- Effects add polish to pages. They can be added to just about any element or triggered by any element, barring browser limitations.

- Spry Effects

- Grow/Shrink- Resizes element
- Appear/Fade- Changes the opacity of an element
- Squish- pushes the element into the top left
- Shake- shakes the element back and forth
- Slide- Moves the element up or down to show/hide
- Blinds- show and hide by covering the element
- Highlight- momentarily changes the color of an element



## Putting it all together



- Use data, widget and effect together.

1. Go to the data table page we made at the beginning. We will put the accordion and blind effect onto the page and put detail regions within it.
2. From the accordion page, copy the markup and constructor script and copy it to the data table page. Also, copy the <script> tag for SpryEffects.js and Accordion.js and paste it into our data page.
3. Position the accordion as necessary.
4. Move the data references from the detailregion to the accordion content panels.
5. To each content div that has spry data, add a spry:detailregion.
6. Preview in Browser.



## Using Spry Effects



1. Open a new page.
2. Use the Draw Layer object to put a layer onto the page. Using the PI, give it a background color.
3. Add some text to the page. We will use this text to control the effect.
4. Make this text a link by highlighting it and putting a # in the PI Link field.
5. To this <a> tag, add the following code:

```
onclick="Spry.Effect.AppearFade('ap1',{duration:1000,from:100,to:20,toggle:true});"
```

  - Within the parens are the following parameters:
  - ('div for effect',duration in milliseconds, start opacity, end opacity, toggle effect:true or false)

**Let's try another one.**

1. Open the accordion page we just made.
2. Add text to the page (not in the accordion): 'Show Details'.
3. Make it a link by selecting the text and adding a # to the Link field in the PI.



## Spry Data, cont.



Now that we have a good overview of the 3 pillars of Spry, we are going to delve into data a bit more.

### Loading data sets

Spry can use multiple data sets and we can use Spry mechanisms to load that new data.

1. On the data table page, add the following code:

```
<a href="#" onclick="ds1.setURL('data/mercury.xml'); ds1.LoadData();" >Load Mercury</a>
```

This example uses 2 functions. The first, setURL, changes the URL to the XML file. The second loads the new XML file. The page automatically updates.

### Spry Conditionals

Spry supports conditional statements with the spry:if attributes. This allow for standard programmatic logic to control output.




## Spry conditionals

- We can use `spry:if` in our example to only display items if they meet the if...then requirements.
 

```
<span spry:if="{good}" == 'yes'">Success</span>
<span spry:if="{good}" == 'no'">Failure</span>
```
- Spry:content**

Spry content is used when you wish to have content markup on the page, but have it replaced with XML data. This is helpful for SEO, because you have static content on the page, but can also XML for a more dynamic page. This is also helpful for JavaScript degradability and accessibility. Let's use `spry:content` in the accordion.

  - In one of the accordion content DIVs, add some descriptive text and wrap it in a `<p tag>`.

2006 Adobe Systems Incorporated. All Rights Reserved. 

## More Spry Attributes

- Spry Content**


`Spry:content` is designed to assist with JavaScript degradability and SEO. It allows you to have static content, and then the content is replaced with the `spry:content` info.

To see it in action, let's go to our main Spry page.

In the accordion, we have dynamic text references in the first panel. We will replace the description text with static text and use `spry:content` to populate the panel.

  - Find the `{description}` reference in the accordion.
  - Replace that reference with dummy text:
 

```
"This is static text."
```

2006 Adobe Systems Incorporated. All Rights Reserved. 22 

## Spry:content

- To this `<p>` tag, add:
 

```
<p spry:content="{desc}">
```


 This will replace your descriptive text with the description from the data set.
- Preview in browser. It should work. Check the browser source code. Your static text is there but the XML data is actually shown.

**Built-in Data References**

Spry has built in data references, that are always available for use. These are helpful in exposing the current state of the data set, like number of records and current record number. We will add these data references to the bottom of the table.

- To the space below the table, and outside the `spry:region` DIV, add the following:
 

```
Record {ds_RowNumberPlus1} of {ds_RowCount}
```

2006 Adobe Systems Incorporated. All Rights Reserved. 

## Spry Content, cont.

- To the container tag `<p>`, add a `Spry:content` attribute. Use the `{description}` data reference as the value of the attribute:
 

```
<p spry:content="{description}">This is dummy text.</p>
```


`Spry:content` can take data references, basic text, or markup.

**Spry States**

Spry has the concept of data states. This allows us to have the page react while working on the data. This is how we can show the "loading" icon while the data loads. If there is an error with the data, we can show that too, without having the page look incorrect.

Let's set up our data table to use spry states.

  - To the `spry:region` that contains our table, add another `<div>`

2006 Adobe Systems Incorporated. All Rights Reserved. 


## Spry Data References

- Wrap this code in a `span` (or `<p>`; it doesn't matter) tag with a `spry:detailregion`. This will cause the data to update when a row is clicked.
- ```
<span spry:detailregion="ds1">Record {ds_RowNumberPlus1} of {ds_RowCount}</span>
```

  - Note:** You just used the oddly named data reference 'ds\_RowNumberPlus1'. We have that because JavaScript uses a zero based counting system. This would cause the first row to display as Record 0. Using this data reference will show Record 1 instead. This can also be used to easily step through the data set.

**Other data references:**

- `ds_RowID` - This is the id of a row in the data set. This id can be used to refer to a specific record in the data set. It does not change even when the data is sorted.
- `ds_RowNumber` - This is the row number of the "current row" of the data set. Within a loop construct, this number reflects the position of the row currently being evaluated.
- `ds_RowNumberPlus1` - This is the same as `ds_RowNumber` except that the first row starts at index 1 instead of index 0.
- `ds_RowCount` - This is the number of rows in the data set. If there is a non-destructive filter set on the data set, this is the total number of rows after the filter is applied.
- `ds_InitialRowNumber` - This is the number of rows in the data set before any non-destructive filter is applied.
- `ds_CurrentRowID` - This is the id of the "current row" of the data set. This value will not change, even when used within a looping construct.
- `ds_CurrentRowNumber` - This is the row number of the "current row" of the data set. This value will not change, even when used within a looping construct.
- `ds_SortColumn` - This is the name of the column last used for sorting. If the data in the data set has never been sorted, this will output nothing (an empty string).
- `ds_SortOrder` - This is the current sort order of the data in the data set. This data reference will output the words "ascending", "descending", or nothing (an empty string).
- `ds_EvenOddRow` - This looks at the current value of `ds_RowNumber` and returns the string "even" or "odd". This is useful for rendering alternate row colors.

2006 Adobe Systems Incorporated. All Rights Reserved. 

## Spry States, cont.

- Add a `spry:state` attribute to the div. The value is 'loading'.
 


```
<div spry:state="loading"></div>
```
- Included in the images folder is a loading gif. Add that to the `<div>`

```
<div spry:state="loading"></div>
```

This content will display while the data set is loading. We need to hide the table until the data is loaded.
- To the `<table>` tag, add another `spry:state` attribute with a 'ready' value:
 

```
<table spry:state="ready">
```

Once the data is loaded, the loading div will go away and be replaced with the ready state content. In our example, it will be brief, but it is more helpful with large data sets or data sets that might be generated from a data base.

2006 Adobe Systems Incorporated. All Rights Reserved. 

## Spry and validation



- Spry uses custom attributes to enable its functionality. This can cause issues for those who need their pages to validate properly. There are a couple ways to handle this.
  - XML namespace- Adding a namespace declaration will ensure that the spry attributes are correctly namespaced.
    - Add `xmlns:spry=http://ns.adobe.com/spry` to the HTML tag.
  - Custom DTD - We wrote a custom DTD extension that will allow Spry pages to validate against the W3C schema.



Better by Adobe.™



## Debugging Spry pages



- Spry has some nice functions that enable you to see what Spry is doing under the browser hood. This is very helpful in debugging your code.
  - Show Spry Region Output  
`Spry.Data.Region.debug = true;`
  - Add this code snippet in a script block under the data set declaration.
  - Preview in browser. It will show 2 things:
    1. The template markup for the region: This is the HTML and Spry in the region.
    2. The generated markup: This is the final displayed HTML for the region.
  - Built in debugging
    - Spry will let you know when the spry markup is incorrect.
      1. Go to your spry page.
      2. In one of the spry:regions, misspell the data set name.
      3. Preview in browser. You should see a warning in the lower right with error feedback.



## Spry Resources



- <http://labs.adobe.com>
- [Spry Forums](#)
- [Spry Samples](#)

