

More XHTML Forms - Custom Forms and The Uni-Form Custom Tag Library

Posted At : December 30, 2007 2:58PM | Posted By : Matt Quackenbush

Related Categories: ColdFusion, Uni-Form Tag Library

In my original post [releasing the tag library](#) I said that I'd post more as I had time to do so. I figured that today I'd take a short break from what I've been working on and write something up.

First, I should point out that there have been a couple of changes made, so the [download](#) has been updated. Be sure to get the latest files. The changes mostly involve textareas: (a) I had forgotten to output the provided value, and (b) I updated the fieldset.cfm and field.cfm files to trim() any content generated by the tags. (The trim() is not mission-critical, just cleans up the resultant HTML a little bit.)

So, without further ado, let's get to it.

Last night I was working on a "special case" form, where I wanted to utilize the tags (since they make my form-writing life so much easier), but needed some custom output to mix in with the form. At first I felt like I had been hit by a Mack truck:

"OMG!" I thought, "I can't use my tags to build this form because they aren't setup to handle this sort of thing! Holy &#^@!! I'm going to have to edit these SOBs so that they can handle custom stuff!"

So I sat there and stared at [CFEclipse](#) for several minutes, as though it would magically rewrite the tags for me, all the while over-analyzing the situation in my mind. (Over-analyzing is my thing. If you need help with mastering the skill, there is no better teacher than yours truly.) And then out of nowhere, that Mack truck really did run me over...

"DOH! You moron! Nothing needs to be rewritten. Just add in your custom code. Frequin ID10T!"

Okay, so it wasn't really a Mack truck...

"Quack, are you going to tell us about it, or are you going to type words all day long?" you interrupt, impatiently. "We want code, dammit! Enough words already, just show us the code!"

The Scenario

As part of this form, there is a *very* long list of checkboxes that need to be included, allowing the user to select from a variety of options. As-is, the tags will only display them either all run together, or each on its own separate line. Instead, I wanted them to be displayed in 3 columns, nice and orderly, and without requiring the user to scroll for half a century to view them all.

The Solution

The options are available to the form in a query. So the view file takes the query, makes preparations for setting up 3 columns, and then displays it. Here's a couple of snippets...

```
.. <cfscript>
}. // break the checkbox options up into 3 columns
}. col1 = ceiling(myQuery.recordCount/3);
}. col2 = col1;
}. col3 = myQuery.recordCount-(col1+col2);
```

```

}.
```

`</cfscript>`

```

7. <div class= "formContainer" >
}.
```

`<!-- build the form -->`

```

}.
```

`<uform:form action= "#myAction#" id= "myForm"`

```

}.
```

`errors= "#myErrors#"
.. submitValue= "#btnSubmit#" >`

```

2. <uform:fieldset legend= "Required Fields" >
}.
```

`<cfoutput> <input`

```

4. type= "hidden"
}.
```

`name= "id"`

```

}.
```

`value= "#id#" /> </cfoutput>`

```

7. <uform:field label= "Name" name= "name"
}.
```

`isRequired= " true "`

```

}.
```

`type= "text"`

```

}.
```

`value= "#form.name#"
.. hint= "Your user name." />`

```

2. <uform:field label= "Message" name= "msg"
}.
```

`type= "textarea"`

```

4. value= "#form.msg#"
}.
```

`hint= "Enter your message." />`

```

}.
```

`</uform:fieldset>`

```

7. <uform:fieldset legend= "Select Options"
}.
```

`class= "blockLabels" >`

```

}.
```

`<div class= "ctrlHolder" >`

```

}.
```

`<p class= "label" > Check each that applies. </p>`

```

.. <cfoutput>
```

`<!-- BEGIN: options column #1 -->`

```

}.
```

`<div class= "float-left clear width-33" >`

```

4. <cfloop from= "1"
}.
```

`to= "#coll#"
}.
index= "i" >

```

7. <div>
}.
```

<label for= "myOptions-#i#" >

```

}.
```

<input type= "checkbox"

```

}.
```

name= "myOptions"

```

.. id= "myOptions-#i#"
}.
```

<cfif

```

}.
```

listFindNoCase(form.myOptions,

```

4. myQuery.id) GT 0 >
````

```

}.
```

`checked= "checked" </cfif>`

```

}.
```

`value= "#myQuery.id#" />`

```

7. &nbsp;#myQuery.option_name#
}.
```

`</label>`

```

}.
```

`</div>`

```

}.
```

`</cfloop>`

```

.. </div>
```

`<!-- END: options column #1 -->`

```

}.
```

`<!-- BEGIN: options column #2 -->`

```

4. <div class= "float-left width-33" >


```

}.
```

<cfloop from= "#coll+1#"
}.
```

`to= "#coll+col2#"
7. index= "i" >`

```

}.
```

`<div>`

```

}.
```

`<label for= "myOptions-#i#" >`

```

}.
```

`<input type= "checkbox"`

```

.. name= "myOptions"


```

}.
```

id= "myOptions-#i#"
}.
```

`<cfif`

```

4. listFindNoCase(form.myOptions,


```

}.
```

myQuery.id) GT 0 >

```

```

}
checked= "checked" </cfif>
}
value= "#myQuery.id#" />
}
  #myQuery.option_name#
}
</label>
}
</div>
..
</cfloop>
}
</div>
}
<!--- END: options column #2 --->
}
<!--- BEGIN: options column #3 --->
}
<div class= "float-left width-33" >
}
<cfloop from= "#col1+col2+1#"
}
to= "#myQuery.recordCount#"
}
index= "i" >
}
<div>
}
<label for= "myOptions-#i#" >
..
<input type= "checkbox"
}
name= "myOptions"
}
id= "myOptions-#i#"
}
<cfif
}
listFindNoCase(form.myOptions,
}
myQuery.id) GT 0 >
}
checked= "checked" </cfif>
}
value= "#myQuery.id#" />
}
  #myQuery.option_name#
}
</label>
..
</div>
}
</cfloop>
}
</div>
}
<!--- END: options column #3 --->
}
</cfoutput>
}
</div>
}
</uform:fieldset>
}
</uform:form>
}
</div>

```

Summary

And there you have it. We've successfully combined the speed and ease of using our tag library to do the bulk of the work for us, while still having our customized section. And the resultant XHTML is still 100% XHTML 1.0 Strict compliant! :)