# Placement of the PagingToolbar on a Grid

Posted At : May 22, 2009 12:33 PM | Posted By : Cutter
Related Categories: ExtJS, ColdFusion, Development, ColdFusion 8

This morning I found I was courtesy copied by Ray Camden, on a reply that he was making to a message he had received from his Blog Contact form from Mike Knox. Mike was trying to find out if it was possible to place the PagingToolbar, used in cfgrid, at the top of the grid rather than the bottom. Ray had told Mike that he had Googled it and not found on obvious solution, and that he was CCing me for my feedback.

Like any padawan, it's pretty humbling for me to be asked advice from the Jedi Master. I've been taking Ray's advice on ColdFusion since the 4.x days, so I try to be pretty diligent when he includes me in helping others with their queries.

I've always held that the use of the CF Ajax components are primarily for rapid application prototyping. They are fantastic for putting up some very basic functionality, but when you need more advanced configuration it is then time to dig in, and go straight to Ext JS itself. Mike's question is a prime example of this. We have nearly no control over a cfgrid's configuration prior to render, and the configuration options that we do have (through the cfgrid and cfgridcolumn attributes) barely scratch the surface to what one can do with an Ext Grid. The PagingToolbar is a class object of Ext and, as such, you should be able to add it to any toolbar of a grid: top, bottom, or both. I fired up Eclipse, booted up my CF and Apache, and pulled up the Ext examples from my local copy of the latest 3.0 download. All of the examples that you can see in the Samples & Demos section of the Ext site are given to you in the download of the library, so that you can run them locally and pick them apart. So, my quest was on.

I went to the Paging Grid Example, and examined the source code of the paging.js file. I quickly found the configuration object for the paging toolbar:

```
var pagingBar = new Ext.PagingToolbar({
        pageSize: 25,
        store: store,
        displayInfo: true,
        displayMsg: 'Displaying topics {0} - {1} of {2}',
        emptyMsg: "No topics to display",

        items:[
            '-', {
            pressed: true,
            enableToggle:true,
            text: 'Show Preview',
            cls: 'x-btn-text-icon details',
            toggleHandler: function(btn, pressed){
                var view = grid.getView();
                view.showPreview = pressed;
                view.refresh();
            }
        }]
    });
```

The PagingToolbar instance is then applied to the Grid, placing the object in the 'bbar' (bottom bar) attribute:

```
// paging bar on the bottom
        bbar: pagingBar
```

After seeing this in the online demo, I went to my local (3.0) demo to see if I could change it. The Ext team rewrote the demo for the upcoming 3.0 release, placing the instance initialization directly in the attribute:

```
// paging bar on the bottom
        bbar: new Ext.PagingToolbar({
            pageSize: 25,
            store: store,
            displayInfo: true,
            displayMsg: 'Displaying topics {0} - {1} of {2}',
            emptyMsg: "No topics to display",
            items:[
                '-', {
                pressed: true,
                enableToggle:true,
                text: 'Show Preview',
                cls: 'x-btn-text-icon details',
                toggleHandler: function(btn, pressed){
                    var view = grid.getView();
                    view.showPreview = pressed;
                    view.refresh();
                }
            }]
        })
```

So, I changed 'bbar' to 'tbar' (top bar) and reloaded the page. Success! The PagingToolbar was now in the top toolbar of the Grid, and fully functional. After reviewing all of this, I decided to go a step further. Could you have two separate PagingToolbar configs on the grid? One at the top and one at the bottom? So, I tried this:

```
// paging bar on the bottom
        bbar: new Ext.PagingToolbar({
            pageSize: 25,
            store: store,
            displayInfo: true,
            displayMsg: 'Displaying topics {0} - {1} of {2}',
            emptyMsg: "No topics to display",
            items:[
                '-', {
                pressed: true,
                enableToggle:true,
                text: 'Show Preview',
                cls: 'x-btn-text-icon details',
```

```
                toggleHandler: function(btn, pressed){
                    var view = grid.getView();
                    view.showPreview = pressed;
                    view.refresh();
                }
            }]
        }),
        // paging bar on the top
        tbar: new Ext.PagingToolbar({
            pageSize: 25,
            store: store,
            displayInfo: true,
            displayMsg: 'Displaying topics {0} - {1} of {2}',
            emptyMsg: "No topics to display",
            items:[
                '-', {
                pressed: true,
                enableToggle:true,
                text: 'Show Preview',
                cls: 'x-btn-text-icon details',
                toggleHandler: function(btn, pressed){
                    var view = grid.getView();
                    view.showPreview = pressed;
                    view.refresh();
                }
            }]
        })
```

That's part of what I love about development, I get to play, experiment, have some (geek) fun. After refreshing the page, I now had two toolbars, at both the top and bottom of the page. Both worked perfectly, and stayed in sync during paging. That's what I love about the Jack Slocum and the Ext Team, they think about things like having more than one paging bar and how they would need to stay in sync, and they've already written it in to the library.

And so, on rare occasion, the student has the opportunity to become the master, though I'm sure there's still plenty more I can learn from Master Camden;)