Late Night Rant on the Flash Help Files and the Datagrid

Posted At : January 20, 2006 12:15 AM | Posted By : Mark Kruger
Related Categories: Flash Remoting

So I'm sitting here after a long day of Flash programming and thinking about how much I hate the flash help system. The little window is *annoying* in the extreme. The font size is too small. The interface is counterintuitive. I can't ever find anything - at least not easily. If I select a combo box and hit the little "help" button in the properties window (I really aught to say "hideous" properties window) I would expect there to get help on the combo box - right. Nope. It always says "sorry - no help can be found for this component." But when I do a search for "combobox" it turns out that there is page after page of help. It's sort of like being white trash at a fancy restaurant. I can see all the empty seats but the hostess just keeps telling me, "sorry... we don't seem to have any openings..."

The help files *do* include samples, but they never seem to be samples of anything I can use. Take the datagrid for example. A typical application using a datagrid would load it with a data object for editing and then pass the edits back to a component for updating - right? Ok, where is the example for simply getting a *row* of the current dataProvider? There are about 20 different notes on how to change the color of a cell, how to fire cell edit events, how to get a list of columns for formatting. The help files have 3 pages of information on style formatting for every 1 example of actually working with data.

```
muse.setStyle("faceColor","RedWithFrustration");
```

Anyway, to solve the problem of "working with a row" I use the following approach that tracks edits. When I populate the grid I have developed a convention of sorts. I add an extra "hidden" column of data to the dataProvider. I call it *edit* or *editflag* or whatever. For example:

```
for(i = 0; i < myRs.getLength(); i++) {


         tmp = {   ID: myRS.getItemAt(i).bom_id,
                DTADDED: myRS.getItemAt(i).DtAdded,
                PART_NO: myRS.getItemAt(i).Part_no,
                PART_DESC: myRS.getItemAt(i).Part_Desc,
                UOM: myRS.getItemAt(i).UOM,
                QUANTITY: myRS.getItemAt(i).Quantity,
                PRICE: myRS.getItemAt(i).CostPerUOM,,
                EDIT: 0
                };
         dataArr.push(tmp);
      }
      _root.my_gd.dataProvider = dataArr;
```

Then I create a "cellEdit" listener that is fired whenever a *value in a cell changes*.

```
   // new listener object var bomCellListener:Object = new Object();

bomCellListener.cellEdit = function(eventObject){
```

```
      my_gd.editField(eventObject.itemIndex, "EDIT", 1);
};

my_gd.addEventListener("cellEdit", bomCellListener);
```

This cellEdit listener serves to flag "rows" in my dataset that have been edited. To save the data I might use a polling application, a button, a dialog or whatever, but when it comes time to fire the "save" function I simply loop through the values and extract only the rows that have been edited to send back to the database. In this respect it works very much like the old "cfgrid".

I know there is also a way to bind a table to the grid in a similar way to a .NET control. I have a natural aversion to leaving any database updates to a wizard or "behind-the-scenes" code (remember cfupdate - shudder). I want to write the databases code and completely control the data in and out - so I have avoided this bound approach. Still, I if anyone has a good tutorial on it I'd be willing to repent :)