

Flex Doc Team

Updates, new content, and other helpful information from the Flex Documentation Team.

[« Chart data drill down examples](#) | [Main](#) | [Wish I'd Said That ... »](#)

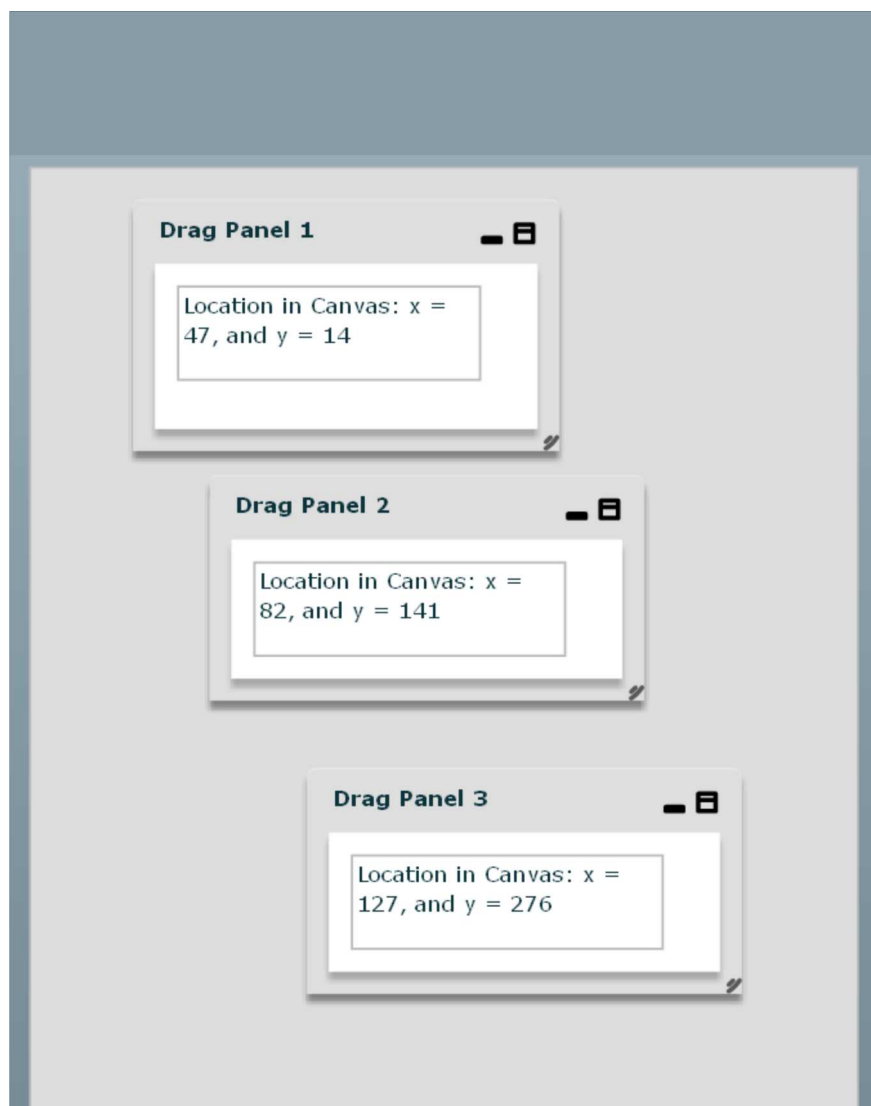
Creating Resizable and Draggable Flex Components

I wanted users of a Flex application to be able to move and resize components at runtime so that the user could lay out the components however they wanted. This application lets the user move, resize, and minimize Panel containers much in the way you can in any windowing environment. While this application uses Panel containers, you could generalize it to for any Flex components.

To use it:

- Click down in the header area of a Panel to drag it around the application. Release the mouse when it is where you want it.
- Click down on the lower-right corner of the Panel to resize it. Release the mouse when it is the size you want. You can not set it to a size smaller than 50 pixels high and 150 pixels wide.
- Click the Minimize/Restore icons to hide/show the body area of the Panel.

The following application shows this functionality:



This application uses three files:

* CanvasDD.mxml - Main app that handles the drag/drop and resizing events. Resizing works for any component that dispatches a 'resizeButtonClicked' event; there is nothing in the Panel needed to resize it except the logic to determine when to dispatch the

COMMENTS

Neat.

The 'x' cursor gets locked on the edge when you drag out of the window.

There's no reaction when the user drags outside the window (when resizing) but does react when dragging the panel.

Otherwise, looks good.

Posted by: [Phillip Kerman](#) | [March 21, 2007 06:25 PM](#)

Very nice work! I just happen to have finished my extension of the Flex Panel component which has similar functionality. [link](#). It doesn't have the rubberband effect but I personally think it's not a mandatory thing. Keep up the great work!

Posted by: [Wietse Veenstra](#) | [March 27, 2007 05:31 AM](#)

This is on the lines what I am looking for. What was your reasoning of not simply placing the Draggable Canvas (which does the dragging) as the parent of the DragPanel class instead of a Panel Object? This way the component will be a little more plug-n-play... I am developing an apollo Dashboard program that has multiple draggable panel sbased on different business metrics (panel for accounting, panel for HR etc) so I need it to be extensible and easily reusable. Any shortcomings you can think of?

Posted by: [David K Roberts](#) | [March 29, 2007 02:53 PM](#)

In the app that I built this prototype for, I wanted to be able to add as many Panels to the Canvas container as necessary. When you create a custom MXML component from a container, and that container has children defined in the custom component, you can only add more children programmatically by calling addChild().

So, if I built a custom Canvas container named MyDragCanvas.mxml as:

```
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml">
<mx:Panel .../>
<mx:Panel .../>
<mx:Panel .../>
</mx:Canvas>
```

When I used in in the application, I could not add more Panels to it in MXML - so this would cause a compiler error:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
<mx:MyDragCanvas>
<!-- You cannot add children in MXML -->
<mx:Panel .../>
<mx:Panel .../>
</mx:MyDragCanvas>
</mx:Application>
```

Stephen

Posted by: [Stephen Gilson](#) | [March 29, 2007 03:39 PM](#)

Okay... back again. I created a mod of your example simplifying it a bit. It kinda deviates from your whole Drag-n-Drop premise and goes in a more mouseevent listener direction... I didn't want to have the outline of the image I was dragging but the actual image, I didn't want to have to add anything to the parent object, and I wanted to easily add multiple dragPanels like any other component without worrying about configuring a main file and alternate .as file...

