

## Re: GridBagLayout and changing constraints objects.

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.gui/2007-03/msg00231.html>

---

- *From:* "Daniel Pitts" <[googlegroupie@xxxxxxxxxxxxxx](mailto:googlegroupie@xxxxxxxxxxxxxx)>
  - *Date:* 25 Mar 2007 23:41:03 -0700
- 

On Mar 25, 10:28 pm, Brandon McCombs <[n...@xxxxxxxx](mailto:n...@xxxxxxxx)> wrote:

Daniel Pitts wrote:

Say I have the following code:

```
JPanel myPanel = new JPanel(new GridBagLayout());
GridBagConstraints constraints = getConstraints();
myPanel.add(new JLabel("Hello world"), constraints);
```

And then later (as a result of an Event), I do something like

```
constraints.gridx = 3;
myPanel.revalidate();
```

Will this have the expected effect, or do I need to remove the old label, and re-add it with the new constraints?

You will have to remove it and re-add it. How is Java to know which Object is supposed to have the new constraint if you don't re-add the Object with that constraint? It could assume to apply that new constraint property to the only Object in a JPanel if there is only one in it but that wouldn't work for more than 1 Object in container and it isn't a good way to do it anyway. It is bad design to need to relocate/resize an Object after an event occurs. You should rethink how you are presenting the interface to the user because the user should not normally see new things appear when they do something; all the Objects in the GUI should be visible, their sizes unchanging, and the unused ones disabled until they are needed (with the enabling trigger being your event).

## Re: GridBagLayout and changing constraints objects.

Actually, I have four objects which would only trade places, and they are similar enough that the change shouldn't bother users. Have you ever played a game on Yahoo games? notice when you sit at a table, your side becomes the southern most side. I'm basically emulating that effect.

It is easier to move the components around than to change which model each component is looking at. Either way, the effect would be the same.

As it is now, the line:

```
myPanel.add(new JLabel("Hello world"), constraints);
```

isn't going to do much because you aren't defining what your constraints are although some of the properties do have default values. The default for gridx is 'relative' which may not be what you want initially.

Hmm, I don't see how you think that my constraints object is unset. I simply didn't copy the code that implemented `getConstraints()`;

.