

## Forum:[Hack Forums](#)

Thread: [JPanel inside a JPanel inside a JScrollPane, not able to scroll](#)

**Tizen** - December 13, 2014, 1:10 pm

I have JPanel wrapped in a JScrollPane, and I'm creating instances my items which use a JPanel, but when I place these into the JPanel wrapped inside a JScrollPane, the JScrollPane won't allow me to scroll.

Image: <https://i.gyazo.com/a276cd66e4ecd3473698b85dec0c4b7c.png>

In more detail; here's my components list:

<https://i.gyazo.com/f7736b88f0a7057659d83d209f9422f1.png>

Now in the JPanel "panel\_servers", I'm placing more instances of JPanel (because each shop item has their own JPanel). And when I do that, scrollPane7 does not allow me to scroll right to see the other shop items.

**Bibl** - December 13, 2014, 1:29 pm

post code pls, i cant diagnose without it

**Cosa Nostra** - December 13, 2014, 3:10 pm

The content that you want scrollable must be on the pane passed to the JScrollPane. Also, it must be populated with content before you pass it or it may have problems ascertaining the dimension. At least I know this was a problem in JDK 1.6, but yeah. Post your code else we won't know for sure.

**MrOrang** - December 13, 2014, 3:18 pm

JPane needs to be Scrollable which is obviously isn't. Look for a container that implements scrollable then add that component to the scrollpane.

Kinda tired right now so I may be wrong but I think that's your problem.

**Cosa Nostra** - December 13, 2014, 3:29 pm

(Dec 13, 2014 04:18 PM)?Looka Wrote: [\[CITE\]](#) JPane needs to be Scrollable which is obviously isn't. Look for a container that implements scrollable then add that component to the scrollpane.

Kinda tired right now so I may be wrong but I think that's your problem.

Correct me if I'm wrong, but I think this is what he wants to do:

<http://pastebin.com/2jmkyGhi>

Which results in a scrollable nested cluster of panels:

My money is still on the fact that when you dynamically add into a scroll panel in java, it's a little problematic because the ScrollPanel logic dictates its vertical and horizontal size at construction and on repaints. There are some methods that allow you to fire that the internal data has changed though, but they're a little complicated and I'd need to see your code first before bothering with an example on that. You could try to just call repaint() on it, but I don't know if that'd do it.

If you want me to help you out and invest time creating a panel that scrolls that you can add and

remove from, let me know and I'll help you out.

Bibl - December 13, 2014, 3:55 pm

(Dec 13, 2014 04:29 PM)Cosa Nostra Wrote: [\[CITE\]](#) Correct me if I'm wrong, but I think this is what he wants to do:

<http://pastebin.com/2jmkyGhi>

Which results in a scrollable nested cluster of panels:

My money is still on the fact that when you dynamically add into a scroll panel in java, it's a little problematic because the ScrollPanel logic dictates its vertical and horizontal size at construction and on repaints. There are some methods that allow you to fire that the internal data has changed though, but they're a little complicated and I'd need to see your code first before bothering with an example on that. You could try to just call repaint() on it, but I don't know if that'd do it.

If you want me to help you out and invest time creating a panel that scrolls that you can add and remove from, let me know and I'll help you out.  
call validate on it

```
Code public static void main(String[] args) throws InterruptedException {
```

```
    // create an instance of the frame
    CalendarFrame frame = new CalendarFrame();
    // display to screen
    frame.setVisible(true);
```

```
    Thread.sleep(1000);
    frame.doit();
```

```
}Code public class CalendarDisplay extends JPanel {
```

```
    private static final long serialVersionUID = 5374598999409020666L;
```

```
    public CalendarDisplay() {
```

```
        // basically setup a box like view, 0=no set limit on the amount of rows, 7=7 columns (7 days of the week)
```

```
        super(new GridLayout(0, 7, 1, 1));
```

```
        Year year = new Year(2000);
```

```
        Month month = Month.FEBRUARY;
```

```
        String[] days = new String[] { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday" };
```

```
        // (days.length * 10) = 10 weeks, change it to however you need it
```

```
        for (int i = 0; i < month.getLength(year); i++) {
```

```
            // [i % 7] divide 0 by 7 and get the remainder(the name of the week)
```

```
            CalendarBox box = new CalendarBox(days[i % 7]);
```

```
            // add this box to the view
```

```
            add(box);
```

```
        }
```

```
    }
```

```
    public void doit() {
```

```
        Year year = new Year(2010);
```

```
Month month = Month.FEBRUARY;
```

```
String[] days = new String[] { "Monday", "Tuesday", "Wednesday", "Thursday", "Friday",  
"Saturday", "Sunday" };  
// (days.length * 10) = 10 weeks, change it to however you need it  
for (int i = 0; i < month.getLength(year); i++) {  
    // [i % 7] divide 0 by 7 and get the remainder(the name of the week)  
    CalendarBox box = new CalendarBox(days[i % 7]);  
    // add this box to the view  
    add(box);  
}  
}  
}Code package calendar.ui;
```

```
import java.awt.BorderLayout;  
import java.awt.Component;  
import java.awt.Container;  
import java.awt.Dimension;
```

```
import javax.swing.JFrame;  
import javax.swing.JMenu;  
import javax.swing.JMenuBar;  
import javax.swing.JMenuItem;  
import javax.swing.JPanel;  
import javax.swing.JScrollPane;  
import javax.swing.ScrollPaneLayout;
```

```
public class CalendarFrame extends JFrame {
```

```
    private static final long serialVersionUID = -2122161377842820073L;
```

```
    public CalendarDisplay calendarPanel;
```

```
    public CalendarFrame() {  
        // set the title; can also call setTitle("Calendar");  
        super("Calendar");  
        // make it so when we press the 'x' button the window and program exits  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        // creates the background jpanel we add everything to  
        JPanel tempPanel = new JPanel(new BorderLayout());  
        // just set the original size of the panel (and in turn the frame, when java dynamically resizes)  
        tempPanel.setSize(850, 450);  
        tempPanel.setPreferredSize(tempPanel.getSize());
```

```
    {  
        // Creates the display panel  
        calendarPanel = new CalendarDisplay();  
        // applies it to a scrollpane so that we can scroll through if there are more days than can be  
displayed  
        JScrollPane scrollPane = new JScrollPane(calendarPanel);  
        // this fixes a wierd bug with the scrollpane overlapping the boxes, dw about it  
        scrollPane.setLayout(new ScrollPaneLayout() {
```

```
private static final long serialVersionUID = -4604183237728799574L;
```

```
@Override
```

```
public Dimension preferredLayoutSize(Container parent) {
```

```
    Dimension dim = super.preferredLayoutSize(parent);
```

```
    JScrollPane pane = (JScrollPane) parent;
```

```
    Component comp = pane.getViewPort().getView();
```

```
    Dimension viewPref = comp.getPreferredSize();
```

```
    Dimension port = pane.getViewPort().getExtentSize();
```

```
    // **Edit 2** changed condition to <= to prevent jumping
```

```
    if (port.height < viewPref.height) {
```

```
        dim.width += pane.getVerticalScrollBar().getPreferredSize().width;
```

```
    }
```

```
    return dim;
```

```
}
```

```
});
```

```
scrollPane.validate();
```

```
scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);
```

```
scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER)
```

```
;
```

```
// put the calendar display to the left (east)
```

```
JPanel bottomPanel = new JPanel(new BorderLayout(10, 10));
```

```
bottomPanel.add(scrollPane, BorderLayout.EAST);
```

```
tempPanel.add(bottomPanel);
```

```
}
```

```
{
```

```
    // Creates the menubar at the top
```

```
JMenuBar menuBar = new JMenuBar();
```

```
    // the file menu
```

```
JMenu fileMenu = new JMenu("File");
```

```
    // sets the f in file to be displayed nicely i think
```

```
fileMenu.setMnemonic('F');
```

```
    // create the 2 menu options
```

```
JMenuItem openItem = new JMenuItem("Open");
```

```
JMenuItem saveItem = new JMenuItem("Save");
```

```
    // add the options to the file menu
```

```
fileMenu.add(openItem);
```

```
fileMenu.add(saveItem);
```

```
    // add the file menu to the main menu bar at the top
```

```
menuBar.add(fileMenu);
```

```
    // set the frames menubar as ours
```

```
setJMenuBar(menuBar);
```

```
}
```

```
// just add all the stuff to the frame; you could alternatively do add(tempPanel); but oh well
```

```
getContentPane().add(tempPanel);
```

```
// basically squeezes the components are resizes them dynamically for us
```

```
pack();
```

```
// put it in the middle of the screen
setLocationRelativeTo(null);
}

public void doit() {
    calendarPanel.doit();
    validate();
}
}[img]http://?.pw/gcb[/img]
```

**Cosa Nostra** - December 13, 2014, 4:00 pm

(Dec 13, 2014 04:55 PM)Bibl Wrote: [\[CITE\]](#) call validate on it  
Ahahaha, I knew there was a method to toggle dataChanged. Nice call. Just like fireDataChanged()  
when dealing with JTable. I got lazy and didn't look it up xD.

**Tizen** - December 13, 2014, 4:26 pm

I've update the OP with more detail on the problem. Thanks!

**Bibl** - December 13, 2014, 4:31 pm

need the code  
you should learn to use the components without a gui builder

**Cosa Nostra** - December 13, 2014, 4:33 pm

I just also remembered, if there is no layout set for the panel that is inside of the scroll pane, the size  
is not dynamically passed to the parent component. So, setting a layout may also help you if you  
have not.

EDIT: Oh, fuck me, you're using a GUI builder? No wonder you're having problems.

**Tizen** - December 13, 2014, 6:58 pm

<https://i.gyazo.com/4bae349398284324c7279e8e4b8d0dea.png>

Managed to get it in the end! Thanks for help guys!