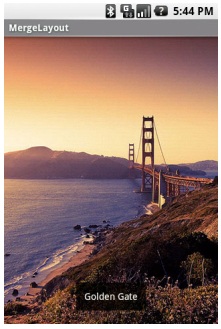
In the previous installment of *Android Layout Tricks*, I showed you how to use the <include /> tag in XML layout to reuse and share your layout code. I also mentioned the <merge /> and it's now time to learn how to use it.

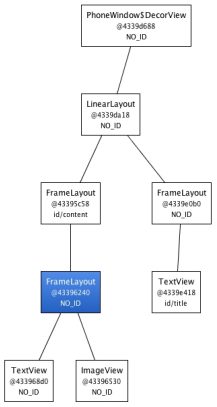
The <merge /> was created for the purpose of optimizing Android layouts by reducing the number of levels in view trees. It's easier to understand the problem this tag solves by looking at an example. The following XML layout declares a layout that shows an image with its title on top of it. The structure is fairly simple; a FrameLayout is used to stack a TextView on top of an ImageView:

```
<FrameLayout xmlns:android=&quot;http://schemas.android.com/apk/res/android&quot;
android:layout_width=&quot;fill_parent&quot; android:layout_height=&quot;fill_parent&quot;>
<ImageView android:layout_width=&quot;fill_parent&quot;
android:layout_height=&quot;fill_parent&quot; android:scaleType=&quot;center&quot;
android:src=&quot;@drawable/golden_gate&quot; /> <TextView
android:layout_width=&quot;wrap_content&quot;
android:layout_height=&quot;wrap_content&quot;
android:layout_marginBottom=&quot;20dip&quot;
android:layout_gravity=&quot;center_horizontal|bottom&quot;
android:padding=&quot;12dip&quot; android:background=&quot;#AA000000&quot;
android:textColor=&quot;#ffffffff&quot; android:text=&quot;Golden Gate&quot;
/> </FrameLayout>
```

This layout renders nicely as we expected and nothing seems wrong with this layout:



Things get more interesting when you inspect the result with HierarchyViewer. If you look closely at the resulting tree you will notice that the FrameLayout defined in our XML file (highlighted in blue below) is the sole child of another FrameLayout:

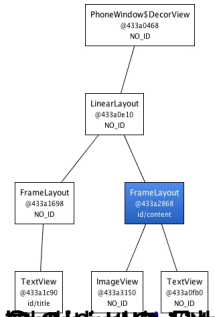


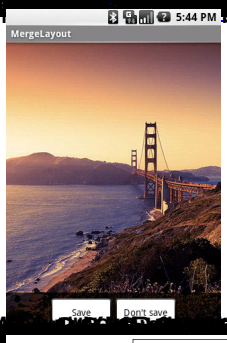
Since our FrameLayout has the same dimension as its parent, by the virtue of using the fill_parent constraints, and does not define any background, extra padding or a gravity, it is *tot ally useless*
. We only made the UI more complex for no good reason. But how could we get rid of this FrameLayout? After all, XML documents require a root tag and tags in XML layouts always represent view instances.

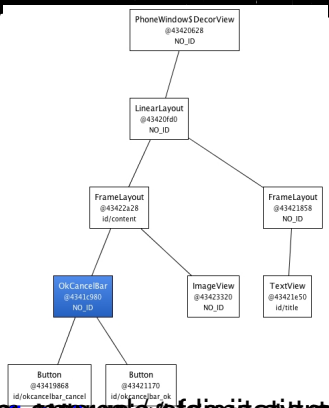
That's where the <merge /> tag comes in handy. When the LayoutInflater encounters this tag, it skips it and adds the <merge /> children to the <merge /> parent. Confused? Let's rewrite our previous XML layout by replacing the FrameLayout with <merge />:

<merge xmlns:android="http://schemas.android.com/apk/res/android"> <ImageView android:layout_width="fill_parent" android:layout_height="fill_parent" android:scaleType="center" android:src="@drawable/golden_gate" /> <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_marginBottom="20dip" android:layout_gravity="center_horizontal|bottom" android:padding="12dip" android:background="#AA000000" android:textColor="#ffffffff" android:text="Golden Gate" /> </merge>

With this new version, both the TextView and the ImageView will be added directly to the top-level FrameLayout. The result will be visually the same but the view hierarchy is simpler:


PhoneWindow$DecorView
@433a0468
NO_ID
LinearLayout
@433a0e10
NO_ID
FrameLayout
@433a1698
NO_ID
FrameLayout
@433a2868
id/content
TextView
@433a1c90
id/title
ImageView
@433a3150
NO_ID
TextView
@433a0fb0
NO_ID


5:44 PM
MergeLayout
Save
Don't save


PhoneWindow$DecorView
@43420628
NO_ID
LinearLayout
@43420f60
NO_ID
FrameLayout
@43422a28
id/content
FrameLayout
@43421858
NO_ID
OkCancelBar
@4341c980
NO_ID
ImageView
@43423320
NO_ID
TextView
@43421e50
id/title
Button
@43419868
id/okcancelbar_cancel
Button
@43421170
id/okcancelbar_ok