ntroduction

Android 4.0, or Ice Cream Sandwich (ICS), introduces GridLayout to support rich user interface design. GridLayout is similar to TableLayout and LinearLayout.

android.widget.GridLayout places its children in a rectangular grid (in the form of rows and columns), similar to the TableLayout.

XML Attributes

GridLayout supports the XML Attributes defined in android.widget.GridLayout class and in android.widget.GridLayout.LayoutParams.

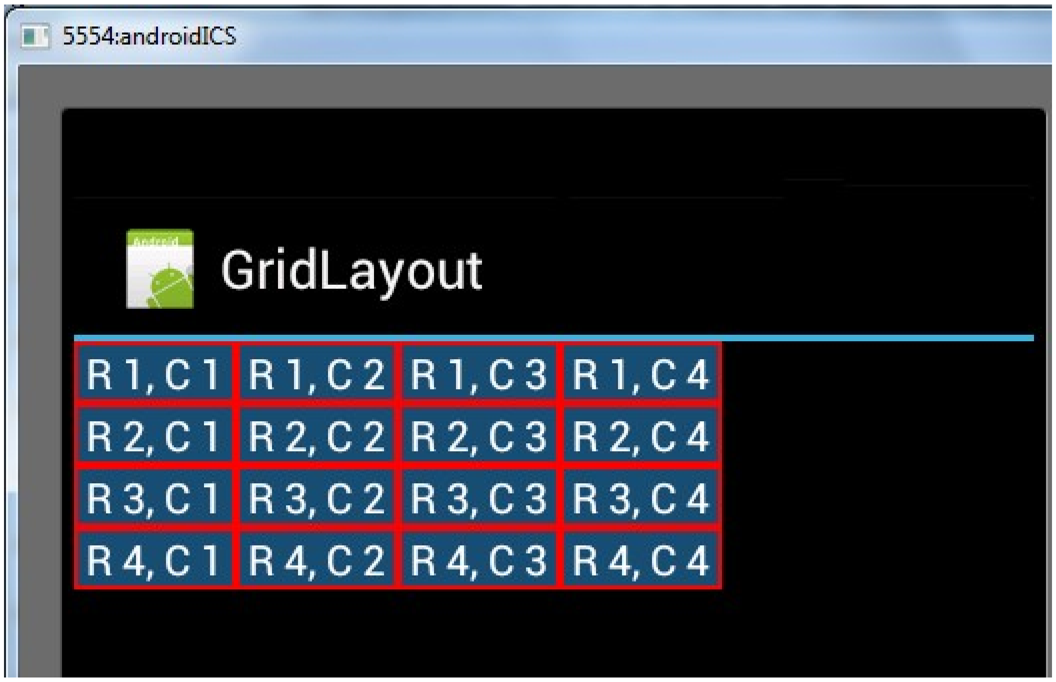
android:rowCount, android:columnCount, android:orientation

Create an xml file with GridLayout of 4 x 4 with xml attributes rowCount and columnCount set to 4 and orientation set as horizontal.

```xml
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:columnCount="4"
  android:rowCount="4"
  android:orientation="horizontal" >
  <TextView android:text=" R 1, C 1 " />
  <TextView android:text=" R 1, C 2 " />
  <TextView android:text=" R 1, C 3 " />
  <TextView android:text=" R 1, C 4 " />
  <TextView android:text=" R 2, C 1 " />
  <TextView android:text=" R 2, C 2 " />
  <TextView android:text=" R 2, C 3 " />
  <TextView android:text=" R 2, C 4 " />
  <TextView android:text=" R 3, C 1 " />
  <TextView android:text=" R 3, C 2 " />
  <TextView android:text=" R 3, C 3 " />
  <TextView android:text=" R 3, C 4 " />
  <TextView android:text=" R 4, C 1 " />
  <TextView android:text=" R 4, C 2 " />
  <TextView android:text=" R 4, C 3 " />
  <TextView android:text=" R 4, C 4 " />
</GridLayout>
```

Code 1: GridLayout Orientation

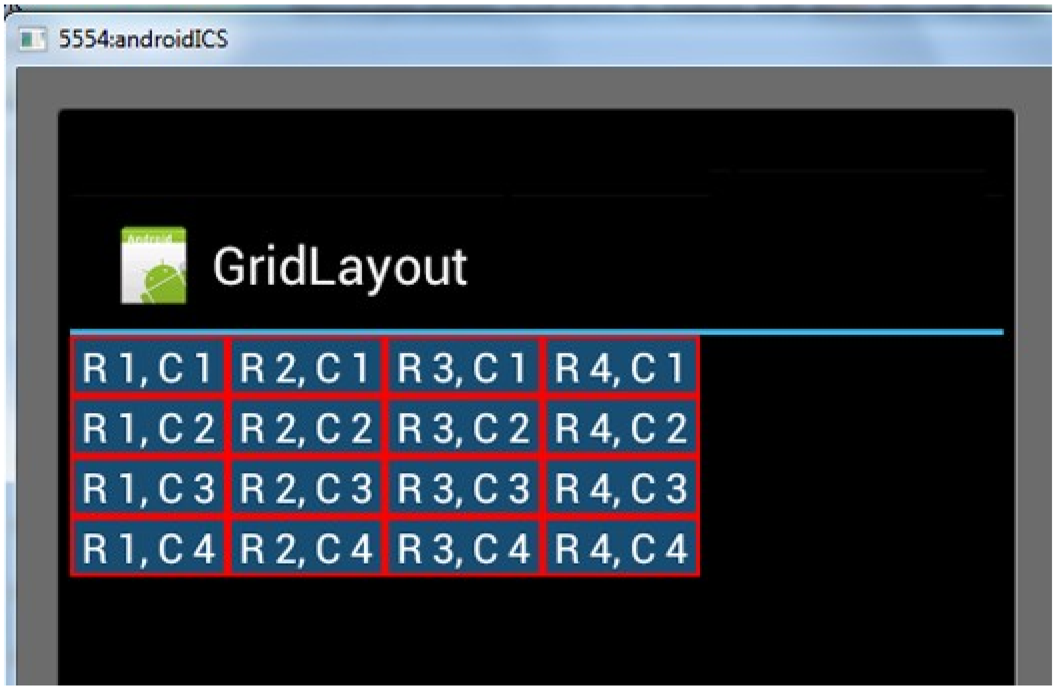
The output is shown in Figure 1. The grid at index 0, that is, “R1 C1” TextView is placed at the leading edge, followed by other grids in sequential manner. The grid at index n will be placed at trailing edge of the container. In this case, GridLayout works like LinearLayout in regards to orientation. The default orientation is landscape, or horizontal.



```
R1,C1 R1,C2 R1,C3 R1,C4
R2,C1 R2,C2 R2,C3 R2,C4
R3,C1 R3,C2 R3,C3 R3,C4
R4,C1 R4,C2 R4,C3 R4,C4
```

Figure 1: GridLayout Orientation Horizontal

With android:orientation="vertical" the above output modifies as shown in Figure 2.



```
R1,C1 R2,C1 R3,C1 R4,C1
R1,C2 R2,C2 R3,C2 R4,C2
R1,C3 R2,C3 R3,C3 R4,C3
R1,C4 R2,C4 R3,C4 R4,C4
```

Figure 2: GridLayout Orientation Vertical

android:layout_gravity

Next, modify the TextView text data placed at R3, C3 to R3, C3 modify. With this the GridLayout output comes out as shown in figure 3.

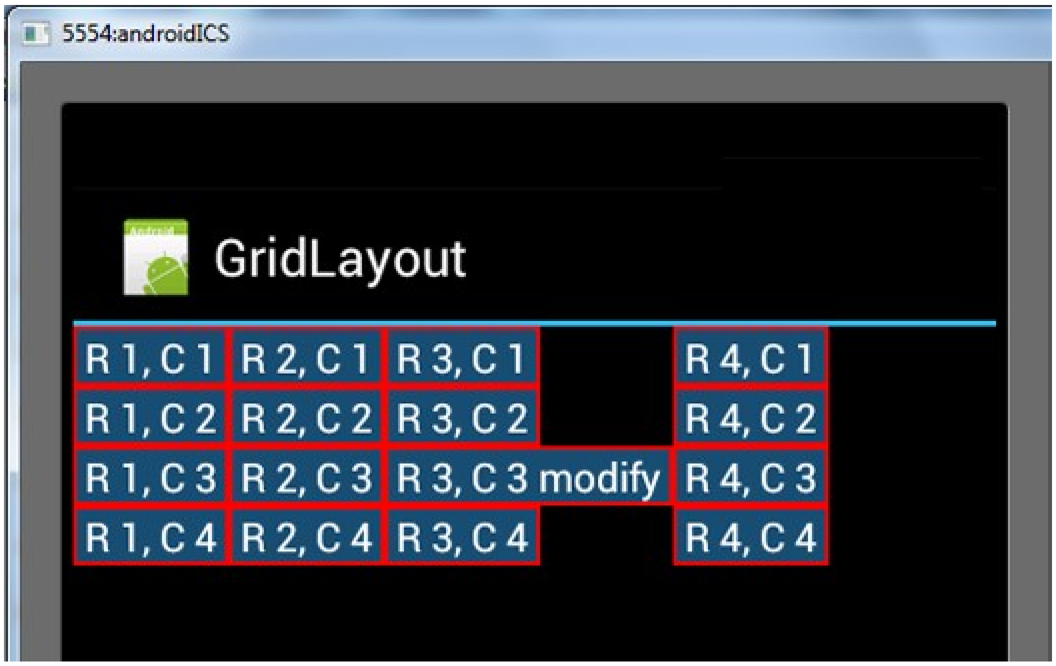


Figure 3: layout_gravity Output

As shown in Figure 3, the cell at R3, C3 has increased its width; however the cells in the C3 columns remain unchanged. This is because in GridLayout, size of each cell is dependent on the layout_gravity of the cell itself and its neighbor cells in same row or column. layout_gravity specifies how a component should be placed in its group of cells. The default value set to layout_gravity is LEFT | BASELINE. For more about the constant values supported by layout_gravity, see the layout_gravity attribute section.
In the following xml code, add android:layout_gravity="fill_horizontal" to each cell in column 3, which produces the output shown in Figure 4.

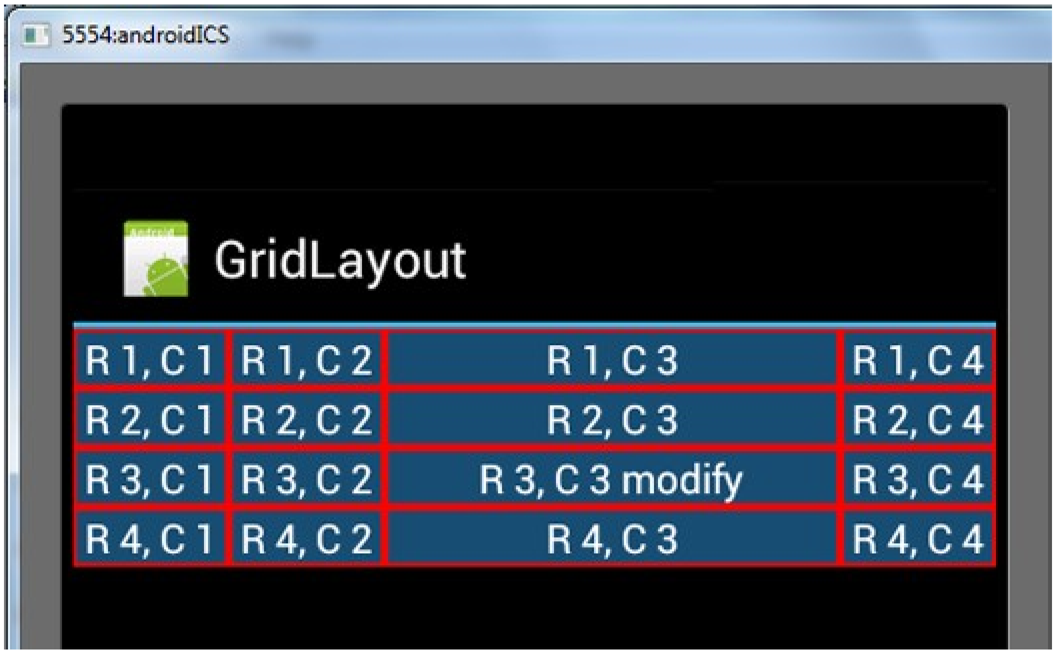
```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:columnCount="4"
  android:rowCount="4" >

  <TextView
    style="@style/textViewStyle"
    android:text=" R 1, C 1 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 1, C 2 " />
  <TextView
    style="@style/textViewStyle"
    android:layout_gravity="fill_horizontal"
    android:text=" R 1, C 3 " />
```

```
<TextView
  style="@style/textViewStyle"
  android:text=" R 1, C 4 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 2, C 1 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 2, C 2 " />
<TextView
  style="@style/textViewStyle"
  android:layout_gravity="fill_horizontal"
  android:text=" R 2, C 3 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 2, C 4 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 3, C 1 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 3, C 2 " />
<TextView
  style="@style/textViewStyle"
  android:layout_gravity="fill_horizontal"
  android:text=" R 3, C 3 modify " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 3, C 4 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 4, C 1 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 4, C 2 " />
<TextView
  style="@style/textViewStyle"
  android:layout_gravity="fill_horizontal"
  android:text=" R 4, C 3 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 4, C 4 " />
```

```
</GridLayout>
```

Code 2: Layout Gravity



R 1, C 1	R 1, C 2	R 1, C 3	R 1, C 4
R 2, C 1	R 2, C 2	R 2, C 3	R 2, C 4
R 3, C 1	R 3, C 2	R 3, C 3 modify	R 3, C 4
R 4, C 1	R 4, C 2	R 4, C 3	R 4, C 4

Figure 4: Output with layout_gravity

android:layout_row, android:layout_column

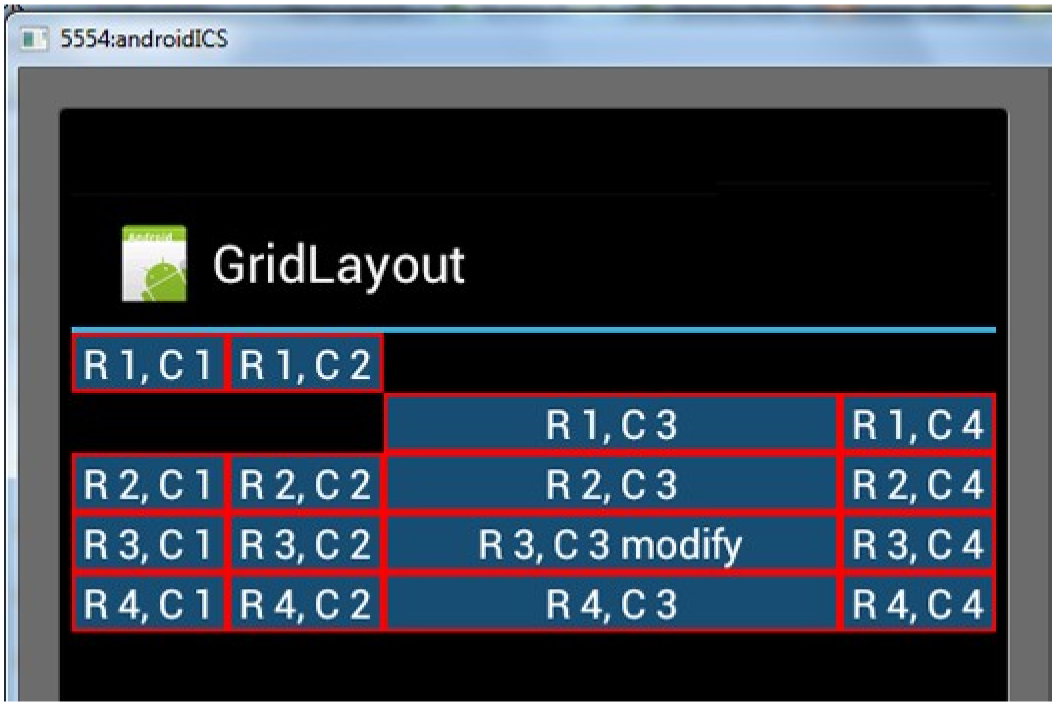
GridLayout can place cells explicitly at particular row and column using android:layout_row and android:layout_column.
Place android:layout_row attribute at cell R1, C3 and see the modify output as shown in figure 5.

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:columnCount="4"
  android:orientation="horizontal"
  android:rowCount="4" >

  <TextView
    style="@style/textViewStyle"
    android:text=" R 1, C 1 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 1, C 2 " />
  <TextView
    style="@style/textViewStyle"
    android:layout_gravity="fill_horizontal"
    android:layout_row="1"
    android:text=" R 1, C 3 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 1, C 4 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 2, C 1 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 2, C 2 " />
  <TextView
    style="@style/textViewStyle"
    android:layout_gravity="fill_horizontal"
    android:text=" R 2, C 3 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 2, C 4 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 3, C 1 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 3, C 2 " />
  <TextView
    style="@style/textViewStyle"
    android:layout_gravity="fill_horizontal"
    android:text=" R 3, C 3 modify " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 3, C 4 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 4, C 1 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 4, C 2 " />
  <TextView
    style="@style/textViewStyle"
    android:layout_gravity="fill_horizontal"
    android:text=" R 4, C 3 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 4, C 4 " />

</GridLayout>
```

Code 3: Layout Row



R 1, C 1 R 1, C 2 R 1, C 3 R 1, C 4
R 2, C 1 R 2, C 2 R 2, C 3 R 2, C 4
R 3, C 1 R 3, C 2 R 3, C 3 modify R 3, C 4
R 4, C 1 R 4, C 2 R 4, C 3 R 4, C 4

Figure 5: layout_row output

As shown in Figure 5, gridlayout places all cells from that row and column. Similarly swapping cells in gridlayout along with cell positions, the developer needs to explicitly set the cells that immediately follow those cells.

Swap the cells at R1, C3 with R2, C3. Set the cell positions of both of these cells, and set the position of the next cell that follows each of these cells:

The modified xml and output is shown in Figure 6.

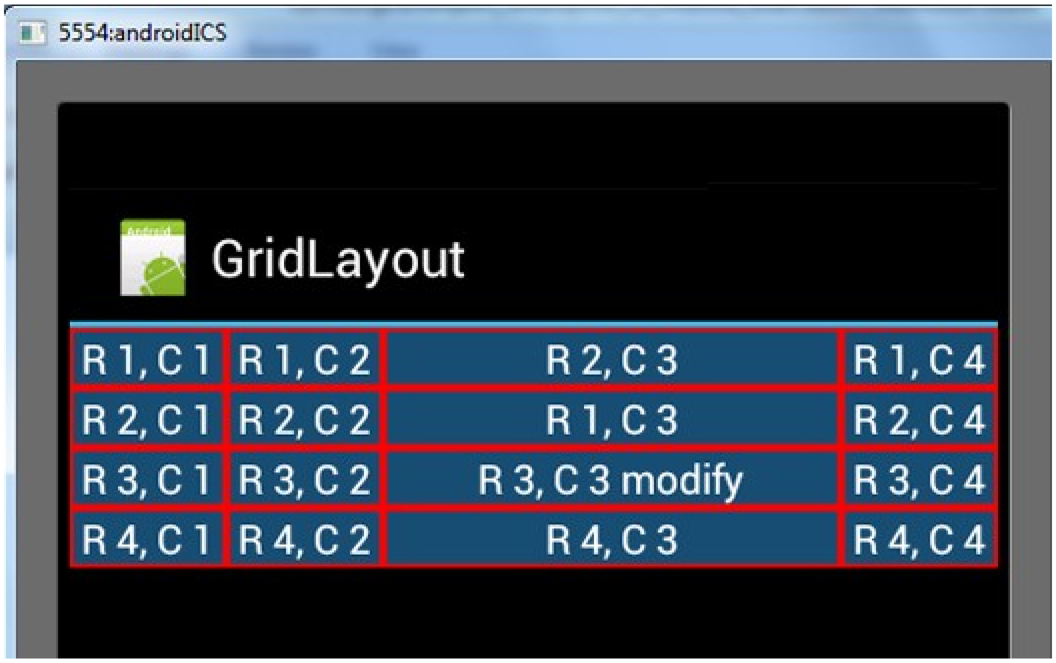
```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:columnCount="4"
  android:orientation="horizontal"
  android:rowCount="4" >

  <TextView
    style="@style/textViewStyle"
    android:text=" R 1, C 1 " />
```

```
<TextView
  style="@style/textViewStyle"
  android:text=" R 1, C 2 " />
<TextView
  style="@style/textViewStyle"
  android:layout_gravity="fill_horizontal"
  android:layout_row="1"
  android:text=" R 1, C 3 " />
<TextView
  style="@style/textViewStyle"
  android:layout_row="0"
  android:text=" R 1, C 4 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 2, C 1 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 2, C 2 " />
<TextView
  style="@style/textViewStyle"
  android:layout_column="2"
  android:layout_gravity="fill_horizontal"
  android:layout_row="0"
  android:text=" R 2, C 3 " />
<TextView
  style="@style/textViewStyle"
  android:layout_row="1"
  android:text=" R 2, C 4 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 3, C 1 " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 3, C 2 " />
<TextView
  style="@style/textViewStyle"
  android:layout_gravity="fill_horizontal"
  android:text=" R 3, C 3 modify " />
<TextView
  style="@style/textViewStyle"
  android:text=" R 3, C 4 " />
<TextView
  style="@style/textViewStyle"
```

```
    android:text=" R 4, C 1 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 4, C 2 " />
  <TextView
    style="@style/textViewStyle"
    android:layout_gravity="fill_horizontal"
    android:text=" R 4, C 3 " />
  <TextView
    style="@style/textViewStyle"
    android:text=" R 4, C 4 " />
</GridLayout>
```

Code 4: Swap Cells



R 1, C 1	R 1, C 2	R 2, C 3	R 1, C 4
R 2, C 1	R 2, C 2	R 1, C 3	R 2, C 4
R 3, C 1	R 3, C 2	R 3, C 3 modify	R 3, C 4
R 4, C 1	R 4, C 2	R 4, C 3	R 4, C 4

Figure 6: layout_row Swap Output

android:layout_rowSpan, android:layout_cloumnSpan

Next, modify the xml used in Figure 4 and remove cell at R2, C3. Now say R1, C3 should occupy the space available after removal of cell R2, C3. This is achieved using the span attribute.

The android:layout_rowSpan, android:layout_columnSpan attribute fills in the place specified by rows and columns. If span attributes are used then android:layout_gravity must be set to “fill”.

The modified xml and output is shown in Figure 7.

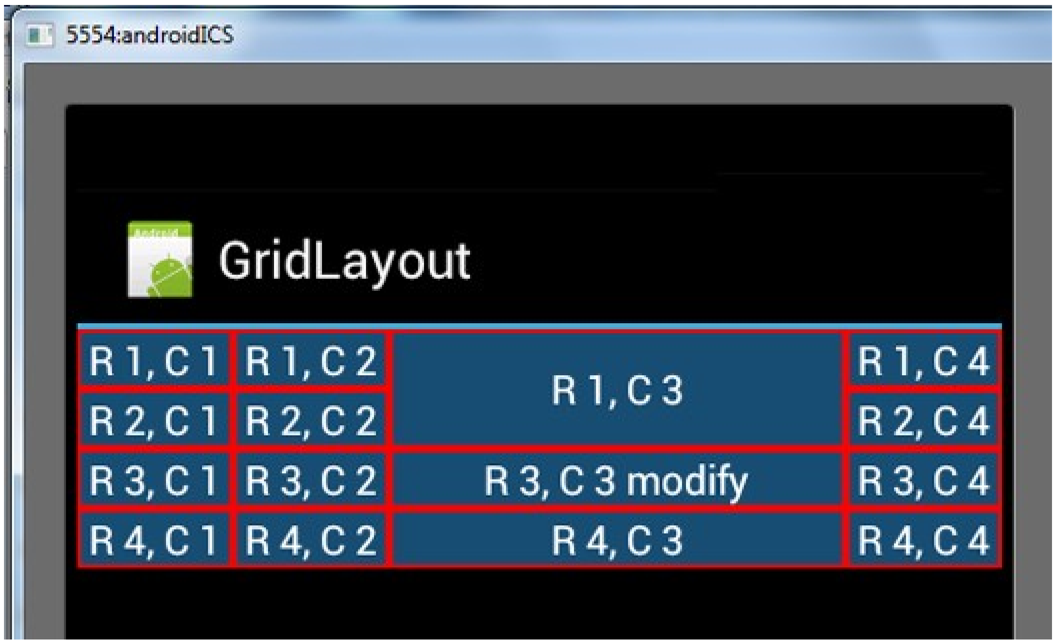
```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="4"
    android:rowCount="4" >

    <TextView
        style="@style/textViewStyle"
        android:text=" R 1, C 1 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 1, C 2 " />
    <TextView
        style="@style/textViewStyle"
        android:layout_gravity="fill"
        android:layout_rowSpan="2"
        android:gravity="center"
        android:text=" R 1, C 3 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 1, C 4 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 2, C 1 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 2, C 2 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 2, C 4 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 3, C 1 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 3, C 2 " />
    <TextView
        style="@style/textViewStyle"
        android:layout_gravity="fill_horizontal"
        android:text=" R 3, C 3 modify " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 3, C 4 " />
    <TextView
```

```
        style="@style/textViewStyle"
        android:text=" R 4, C 1 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 4, C 2 " />
    <TextView
        style="@style/textViewStyle"
        android:layout_gravity="fill_horizontal"
        android:text=" R 4, C 3 " />
    <TextView
        style="@style/textViewStyle"
        android:text=" R 4, C 4 " />
</GridLayout>
```

Code 5:Row Span



R 1, C 1	R 1, C 2	R 1, C 3	R 1, C 4
R 2, C 1	R 2, C 2		R 2, C 4
R 3, C 1	R 3, C 2	R 3, C 3 modify	R 3, C 4
R 4, C 1	R 4, C 2	R 4, C 3	R 4, C 4

Figure 7: Rowspan Output

ref:http://developer.samsung.com/android/technical-docs/GridLayout-in-Android