



media
computing
group



RWTHAACHEN
UNIVERSITY

Android Specifics

Jonathan Diehl (Informatik 10)
Hendrik Thüs (Informatik 9)

Android Specifics

- ArrayAdapter
- Preferences
- Widgets

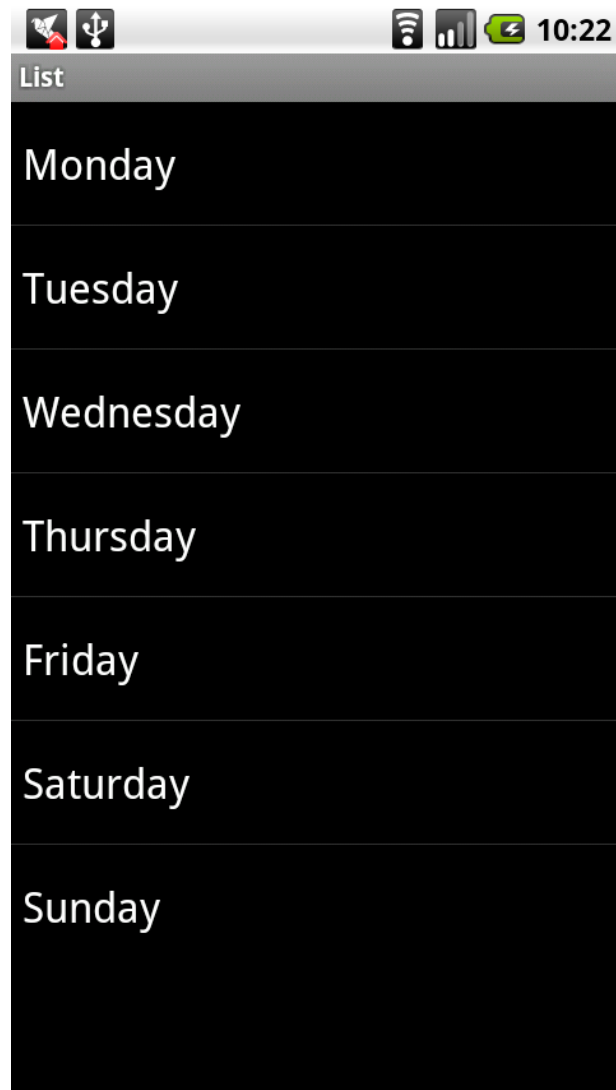
ArrayAdapter

ListView

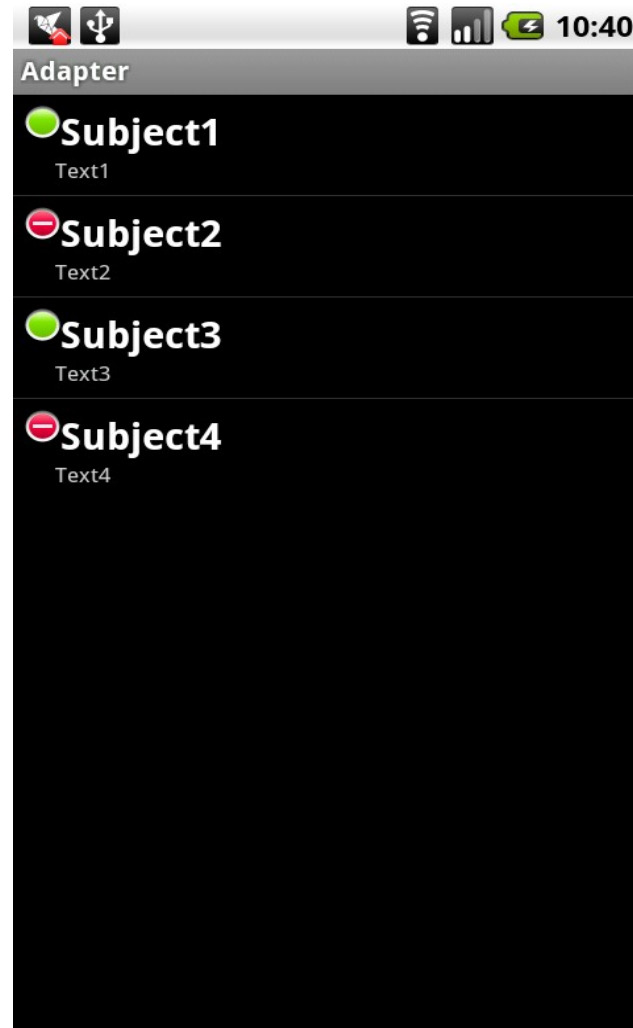
```
<ListView  
    android:layout_height="wrap_content"  
    android:id="@+id/listView1"  
    android:layout_width="fill_parent"  
    android:entries="@array/days" />
```

- Presenting elements, given in an array
- `onClick`Listener
- `onLongClick`Listener

ListView



ArrayAdapter



ArrayAdapter

- (1) Create a ListItem
- (2) Implement an object
- (3) Implement an ArrayAdapter
- (4) Combine everything

ListItem

```
<LinearLayout  
  android:orientation="vertical"  
  android:layout_height="fill_parent"  
  android:layout_weight="1"  
  android:layout_width="wrap_content">
```

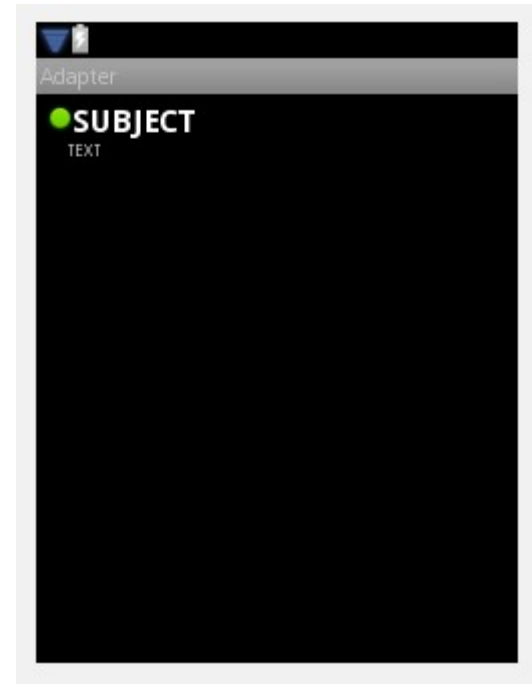
```
<LinearLayout  
  android:id="@+id/LinearLayout09"  
  android:layout_width="fill_parent"  
  android:layout_height="wrap_content"  
  android:orientation="horizontal">
```

```
<ImageView  
  android:id="@+id/itemDone"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:background="@android:drawable/presence_online">  
</ImageView>
```

```
<TextView  
  android:id="@+id/itemSubject"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:text="SUBJECT"  
  android:textStyle="bold"  
  android:textSize="20dp"  
  android:textColor="#ffffff">  
</TextView>  
</LinearLayout>
```

```
<TextView  
  android:id="@+id/itemText"  
  android:layout_width="wrap_content"  
  android:text="TEXT"  
  android:layout_height="wrap_content"  
  android:layout_marginLeft="15dp"  
  android:textSize="11dp"  
  android:textColor="#cdcdcd">  
</TextView>
```

```
</LinearLayout>
```



Object

```
public class Task {  
  
    private String subject;  
    private String text;  
    private Boolean done;  
  
    public Task(String subject, String text, Boolean done) {  
        this.subject = subject;  
        this.text = text;  
        this.done = done;  
    }  
  
    public String getSubject() {  
        return subject;  
    }  
  
    public void setSubject(String subject) {  
        this.subject = subject;  
    }  
  
    public String getText() {  
        return text;  
    }  
  
    public void setText(String text) {  
        this.text = text;  
    }  
  
    public Boolean getDone() {  
        return done;  
    }  
  
    public void setDone(Boolean done) {  
        this.done = done;  
    }  
  
}
```

ArrayAdapter

```
public class TaskOrderAdapter extends ArrayAdapter<Task> {
    private ArrayList<Task> items;
    private Context context;

    public TaskOrderAdapter(Context context, int
        textViewResourceId,
        ArrayList<Task> items) {

        super(context, textViewResourceId, items);
        this.items = items;
        this.context = context;
    }

    public void setItems(ArrayList<Task> items) {
        this.items = items;
    }
}
```

```
@Override
```

```
public View getView(int position, View convertView, ViewGroup parent) {  
    View v = convertView;  
    if (v == null) {  
        LayoutInflater vi = (LayoutInflater) context  
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
        v = vi.inflate(R.layout.tasklistitem, null);  
    }  
    Task task = items.get(position);  
    if (task != null) {  
        [Looking for child views: itemDone, itemSubject, itemText]  
  
        if (itemDone != null) {  
            if (task.getDone()) {  
                itemDone.setBackgroundResource(  
                    android.R.drawable.presence_online);  
            } else {  
                itemDone.setBackgroundResource(  
                    android.R.drawable.presence_busy);  
            }  
        }  
  
        if (itemSubject != null) {  
            itemSubject.setText(task.getSubject());  
        }  
  
        if (itemText != null) {  
            itemText.setText(task.getText());  
        }  
    }  
    return v;  
}
```

Activity

```
public class main extends Activity {

public static TaskOrderAdapter adapter = null;
private ArrayList<Task> taskArray;

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    taskArray = new ArrayList<Task>();

    Task task1 = new Task("Subject1", "Text1", true);
    Task task2 = new Task("Subject2", "Text2", false);
    Task task3 = new Task("Subject3", "Text3", true);
    Task task4 = new Task("Subject4", "Text4", false);

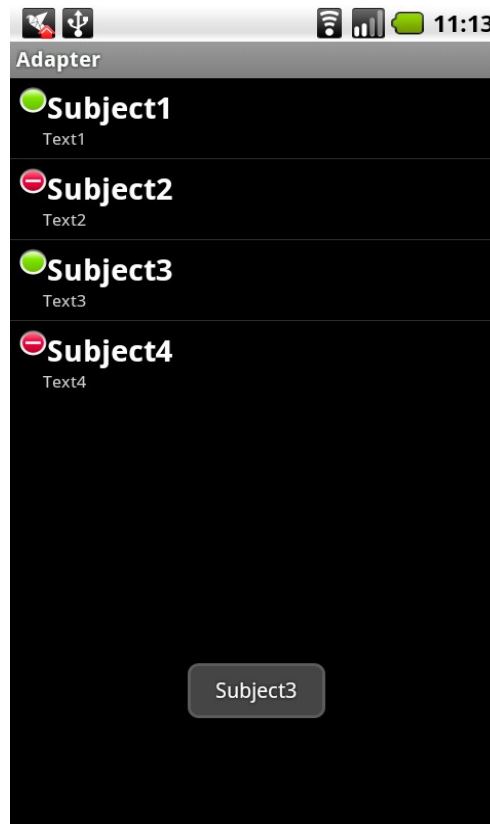
    taskArray.add(task1);
    taskArray.add(task2);
    taskArray.add(task3);
    taskArray.add(task4);

    ListView taskList = (ListView) findViewById(R.id.taskList);

    adapter = new TaskOrderAdapter(main.this,
        R.layout.tasklistitem, taskArray);
    taskList.setAdapter(adapter);
    ...
}
```

onClickListener

```
taskList.setOnItemClickListener(new OnItemClickListener() {  
    public void onItemClick(AdapterView<?> a, View v, int position, long id) {  
        Toast.makeText(getApplicationContext(),  
            taskArray.get(position).getSubject(), Toast.LENGTH_SHORT).show();  
    }  
});
```



ContextMenu

```
registerContextMenu(taskList);
```

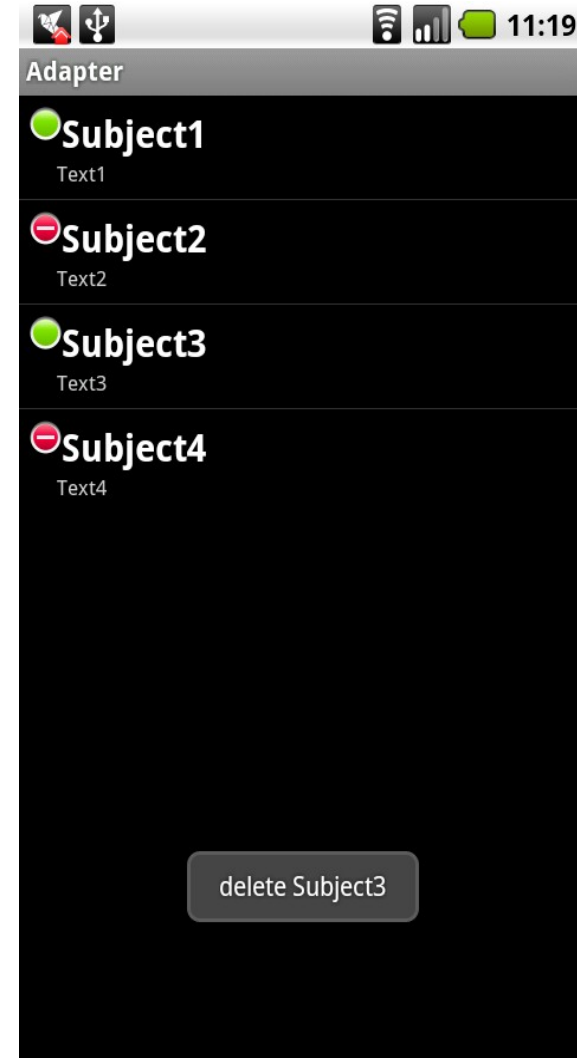
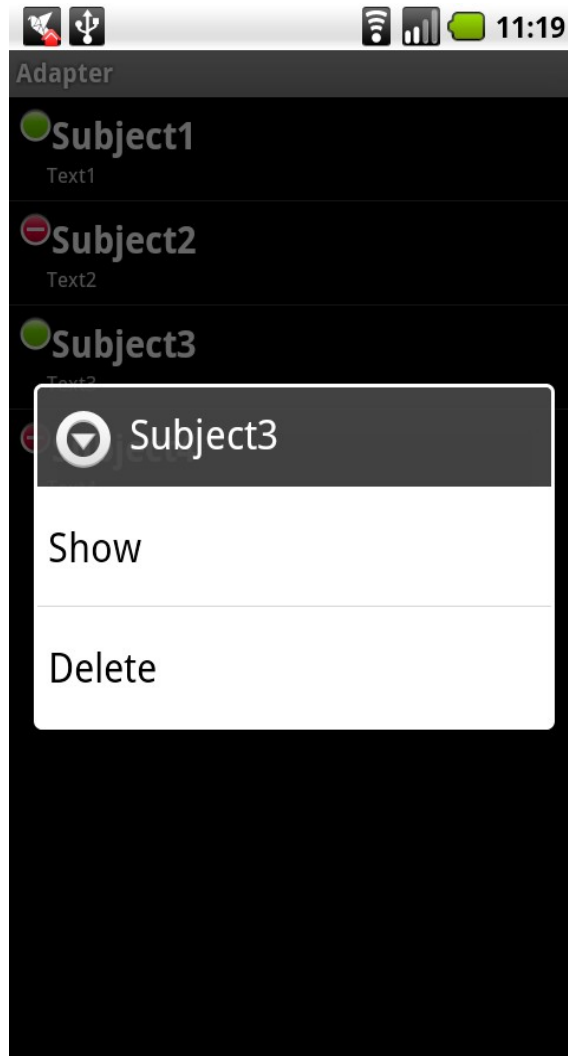
```
@Override
```

```
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {  
    if (v.getId() == R.id.taskList) {  
        AdapterView.AdapterContextMenuInfo info =  
            (AdapterView.AdapterContextMenuInfo) menuInfo;  
        menu.setHeaderTitle(taskArray.get(info.position).getSubject());  
        String[] menuItems = getResources().getStringArray(R.array.tasksContextMenu);  
        for (int i = 0; i < menuItems.length; i++) {  
            menu.add(Menu.NONE, i, i, menuItems[i]);  
        }  
    }  
}
```

```
@Override
```

```
public boolean onContextItemSelected(MenuItem item) {  
    int menuItemIndex = item.getItemId();  
    AdapterView.AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();  
  
    if (menuItemIndex == 0) {  
        Toast.makeText(this, "show " + taskArray.get(info.position).getSubject(),  
            Toast.LENGTH_SHORT).show();  
    } else if (menuItemIndex == 1) {  
        Toast.makeText(this, "delete " + taskArray.get(info.position).getSubject(),  
            Toast.LENGTH_SHORT).show();  
    }  
    return true;  
}
```

Android Specifics

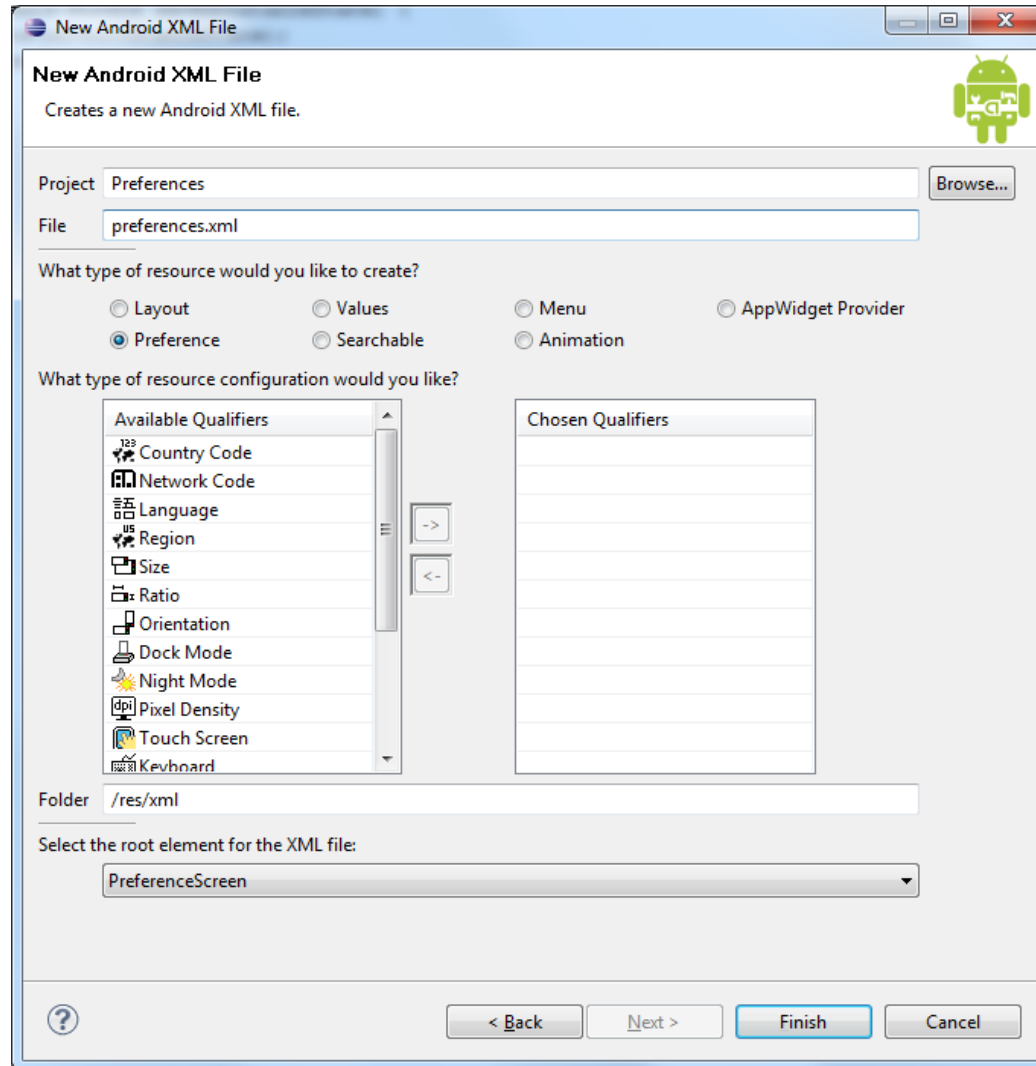


Preferences

Preferences

- Primitive data-sets
- Key-value pairs
- Kept private
- Persists killing the application
- DON'T use SharedPreferences as database or for large data-sets

Xml for Preferences



Xml Content

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
  <PreferenceCategory android:title="category 1">
    <CheckBoxPreference android:title="checkbox 1"
      android:key="item1"></CheckBoxPreference>
    <EditTextPreference android:dependency="item1"
      android:title="edittext1" android:key="item2"></EditTextPreference>
  </PreferenceCategory>
</PreferenceScreen>
```

The screenshot shows the Android Studio interface with a tree view on the left. The tree view contains the following elements:

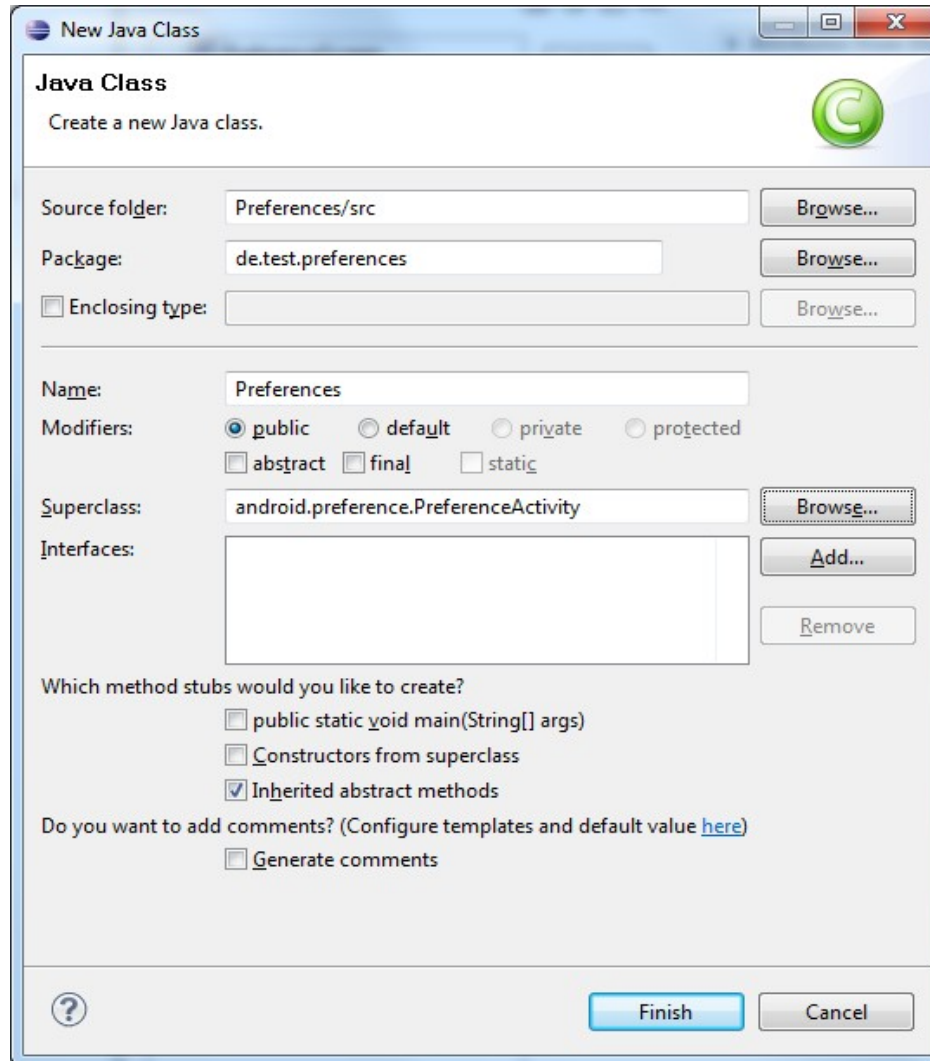
- PreferenceScreen (selected)
- PreferenceCategory (selected)
- item1 (CheckBoxPreference)
- item2 (EditTextPreference)

Below the tree view, there are four buttons: Add..., Remove..., Up, and Down.

The screenshot shows the 'Attributes from Preference' dialog in Android Studio. The dialog has a title bar 'Attributes from Preference' and a section 'Attributes from Preference' with the following fields:

Key	item2	Browse...
Title	edittext1	Browse...
Summary		Browse...
Order		
Layout		Browse...
Widget layout		Browse...
Enabled		
Selectable		
Dependency	item1	Browse...
Persistent		
Default value		Browse...
Should disable view		

PreferenceActivity



PreferenceActivity

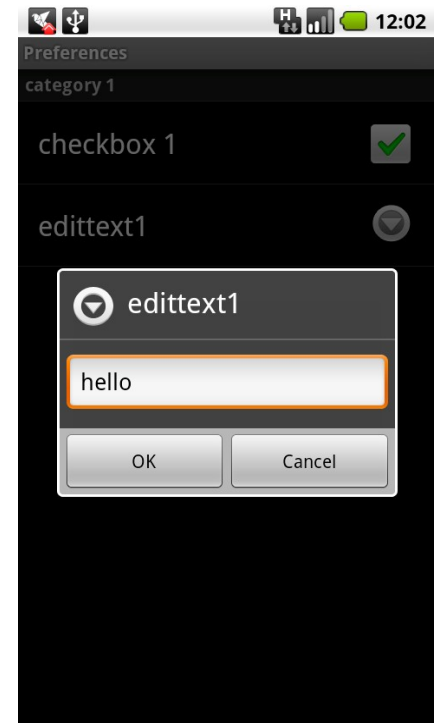
```
public class Preferences extends PreferenceActivity {  
    /** Called when the activity is first created. */  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        addPreferencesFromResource(R.xml.preferences);  
    }  
}
```

Combining everything

```
SharedPreferences preferences;
```

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main);  
    preferences = PreferenceManager.getDefaultSharedPreferences(this);  
}
```

```
i = new Intent(main.this, Preferences.class);  
startActivity(i);
```

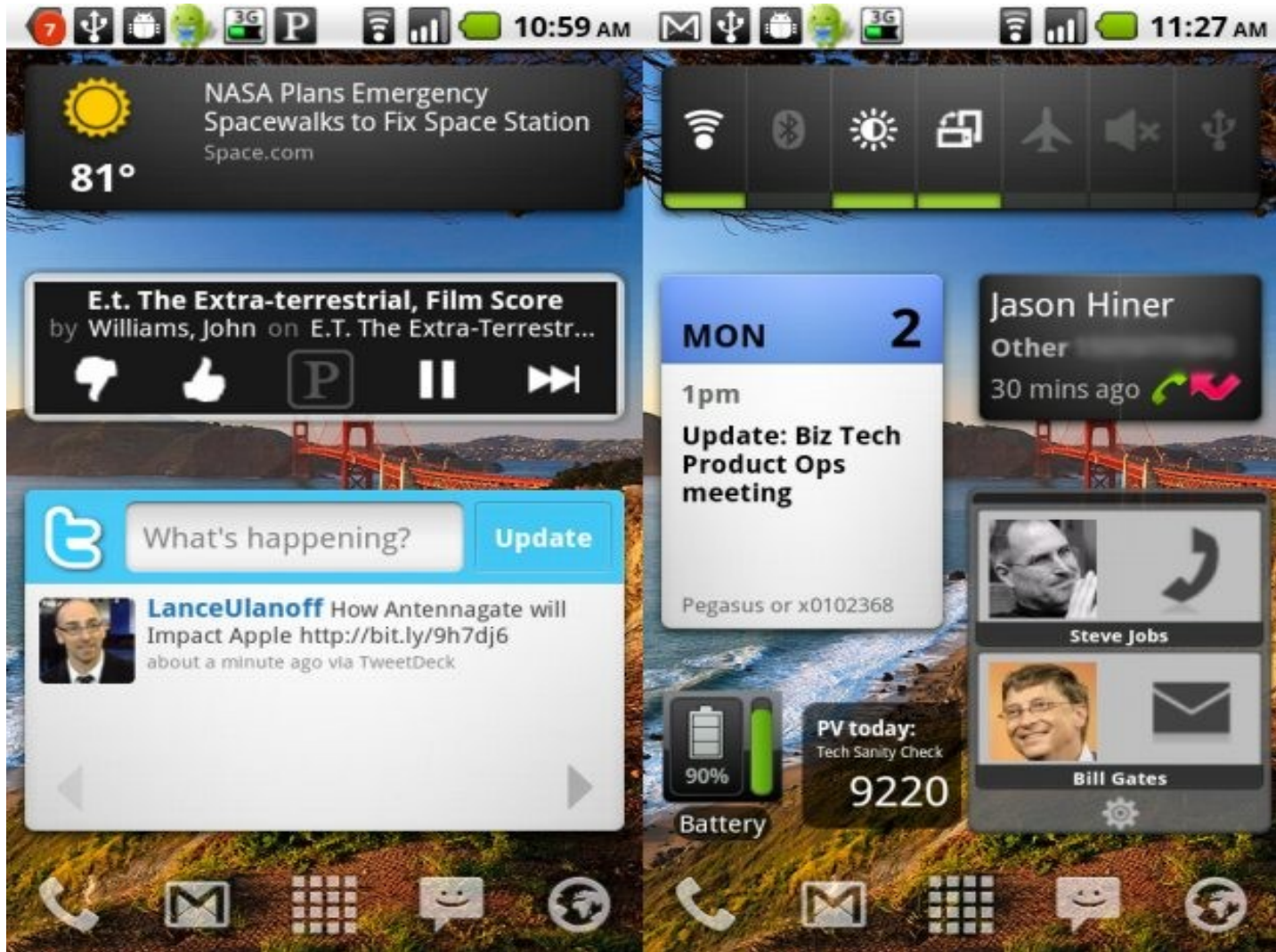


Retrieving Preferences

```
Boolean checkbox1 = preferences.getBoolean("item1", false);  
String edittext1 = preferences.getString("item2", "n/a");
```

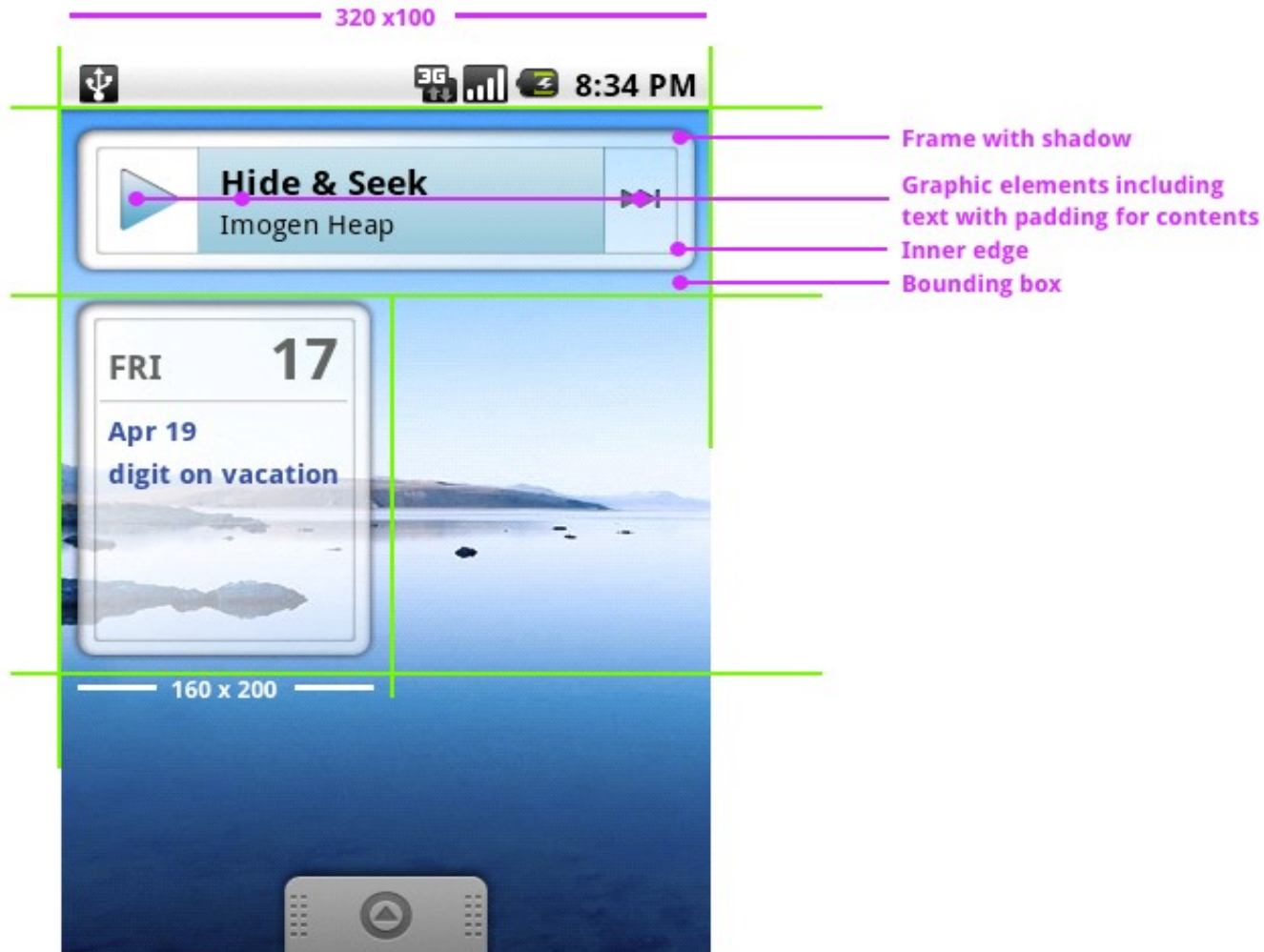
abstract boolean	<code>getBoolean(String key, boolean defValue)</code> Retrieve a boolean value from the preferences.
abstract float	<code>getFloat(String key, float defValue)</code> Retrieve a float value from the preferences.
abstract int	<code>getInt(String key, int defValue)</code> Retrieve an int value from the preferences.
abstract long	<code>getLong(String key, long defValue)</code> Retrieve a long value from the preferences.
abstract String	<code>getString(String key, String defValue)</code> Retrieve a String value from the preferences.

Widgets



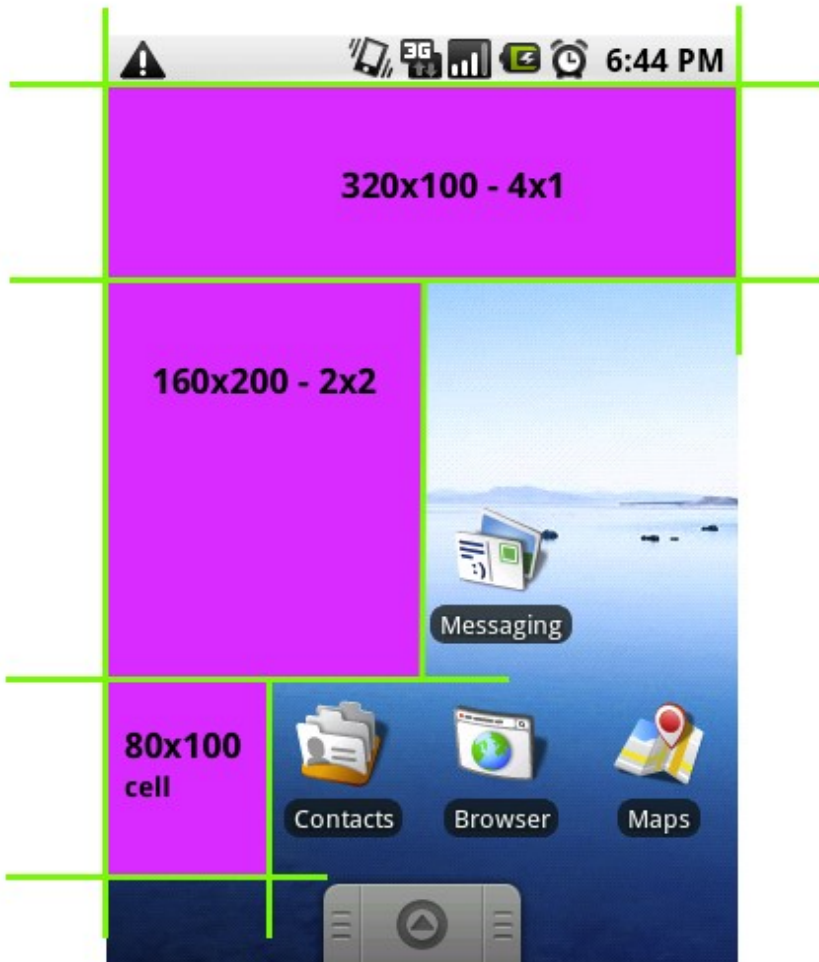
<http://www.techrepublic.com/photos/15-must-have-widgets-for-android/452087>

Design Guidelines



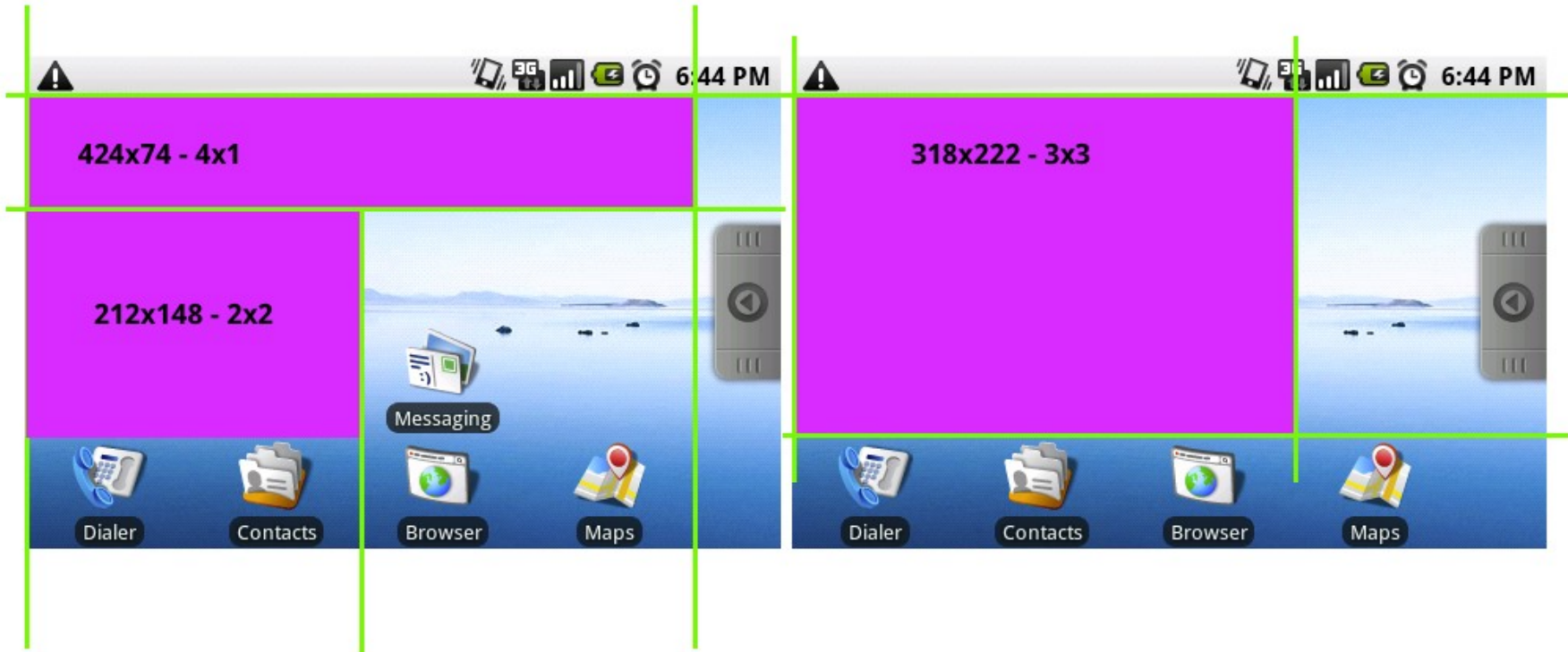
http://developer.android.com/guide/practices/ui_guidelines/widget_design.html

Sizes - Portrait



http://developer.android.com/guide/practices/ui_guidelines/widget_design.html

Sizes - Landscape



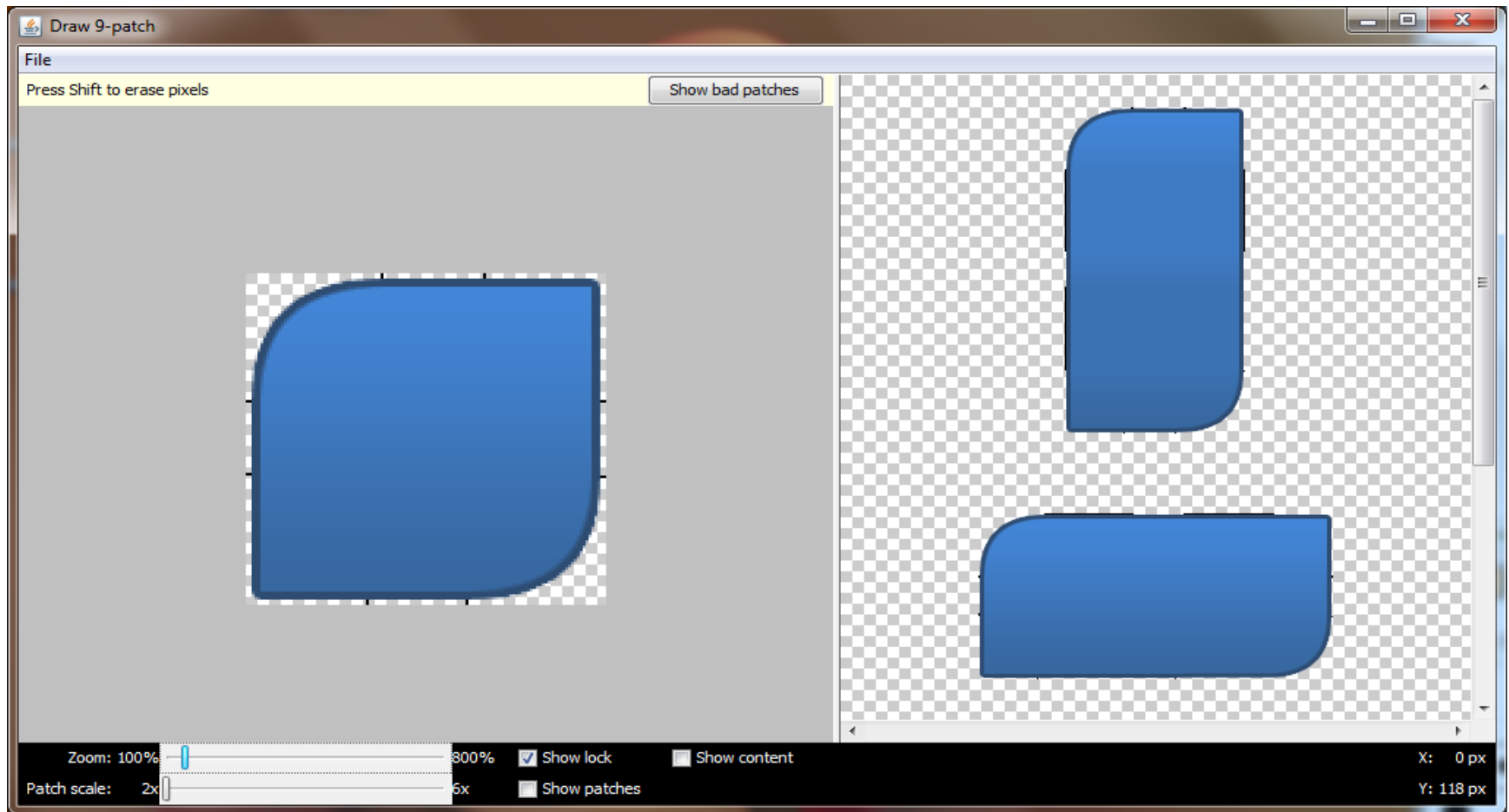
Sizes

- Portrait (per cell):
 - 80 pixels wide
 - 100 pixels tall
- Landscape (per cell):
 - 106 pixels wide
 - 74 pixels tall

Sizes - Worst Case

(number of cells * 74) – 2 dp

9-patch Images

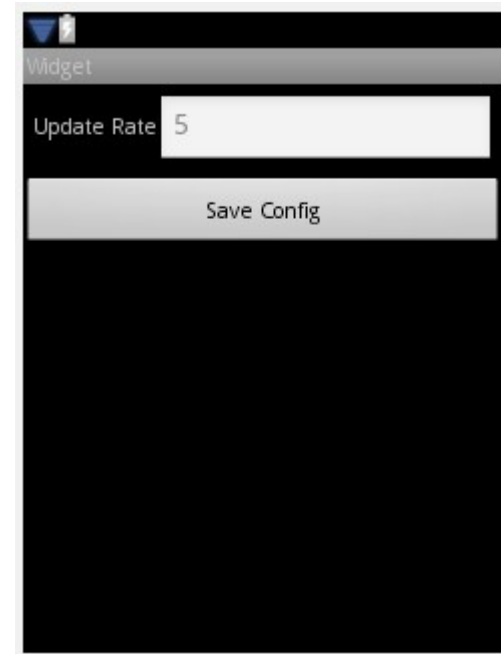
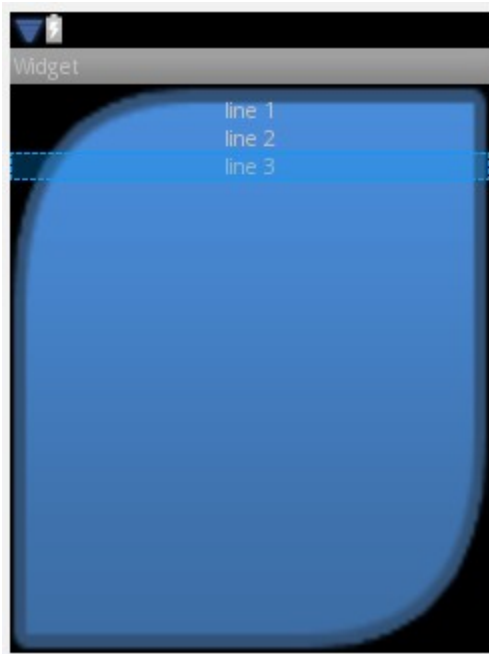


<http://developer.android.com/guide/developing/tools/draw9patch.html>

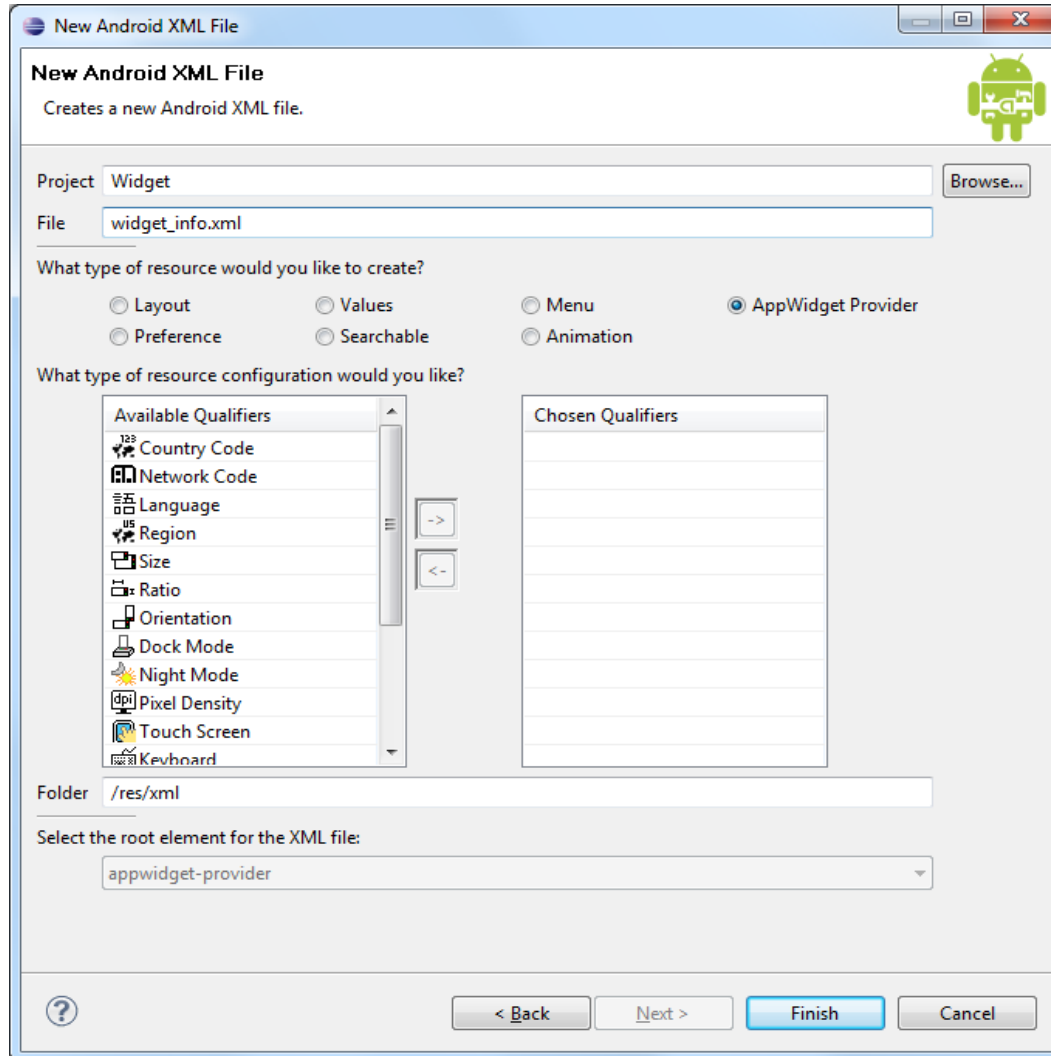
Create an App Widget

- (1) Layout of the Widget
- (2) Layout of the configuration
- (3) AppWidget Provider meta file
- (4) AppWidgetProvider
- (5) Activity to handle configuration

Layouts




AppWidget Provider meta file



AppWidget Provider meta file

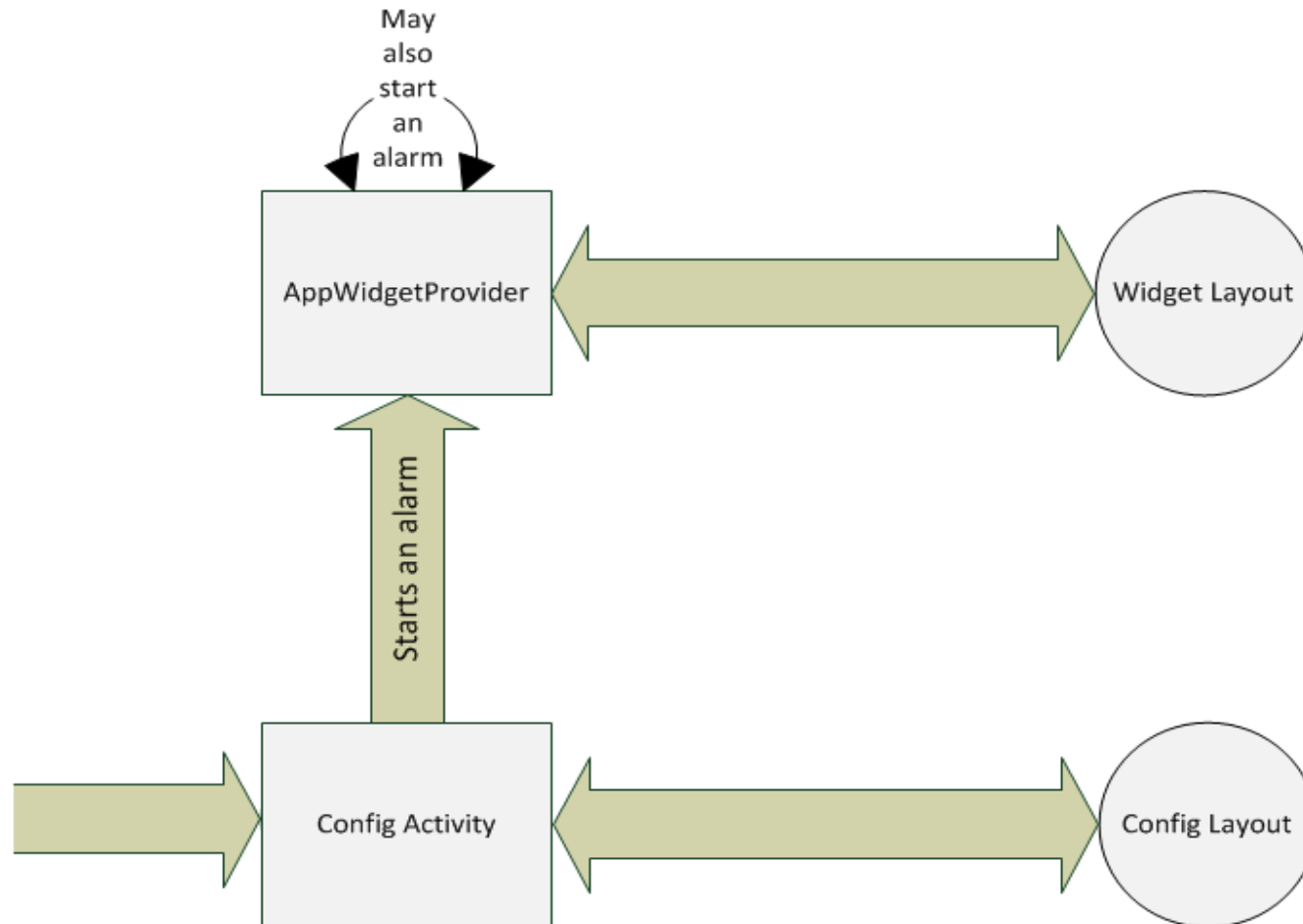
Attributes for AppWidget Provider

 Use [appwidget-provider](#) as the root tag of the XML resource that describes an AppWidget provider.

Min width	<input type="text" value="146dp"/>
Min height	<input type="text" value="72dp"/>
Update period millis	<input type="text"/>
Initial layout	<input type="text" value="@layout/widget"/> <input type="button" value="Browse..."/>
Configure	<input type="text" value="de.test.widget.WidgetConfiguration"/> <input type="button" value="Browse..."/>

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="146dp"
    android:minHeight="72dp"
    android:initialLayout="@layout/widget"
    android:configure="de.test.widget.WidgetConfiguration">
</appwidget-provider>
```

Widget



xcode

What are questions concerning
xcode?

If you think of more, email Jonathan