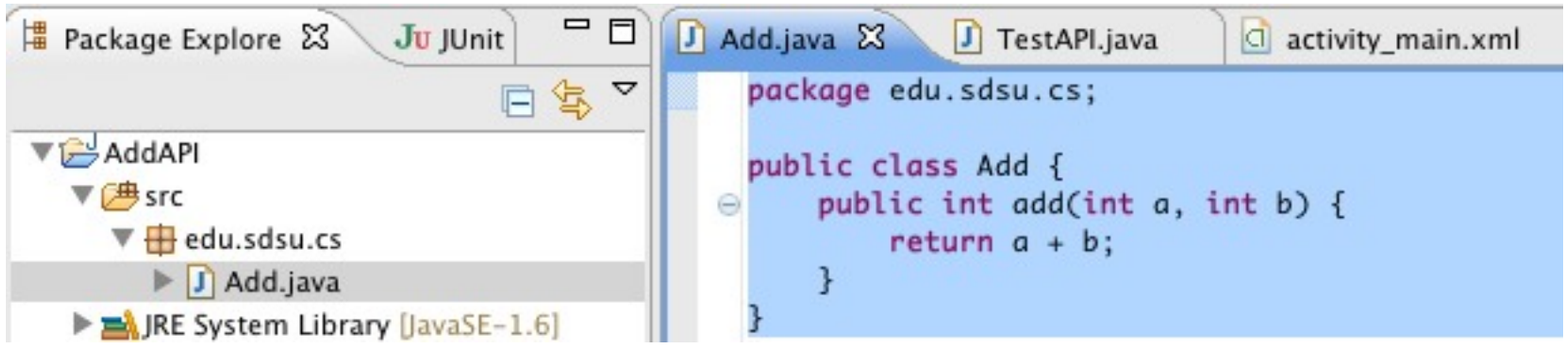
CS 580 Client-Server Programming
Fall Semester, 2012
Doc 11 Android Lists
Oct 2, 2012

Creating & Using
a Jar File

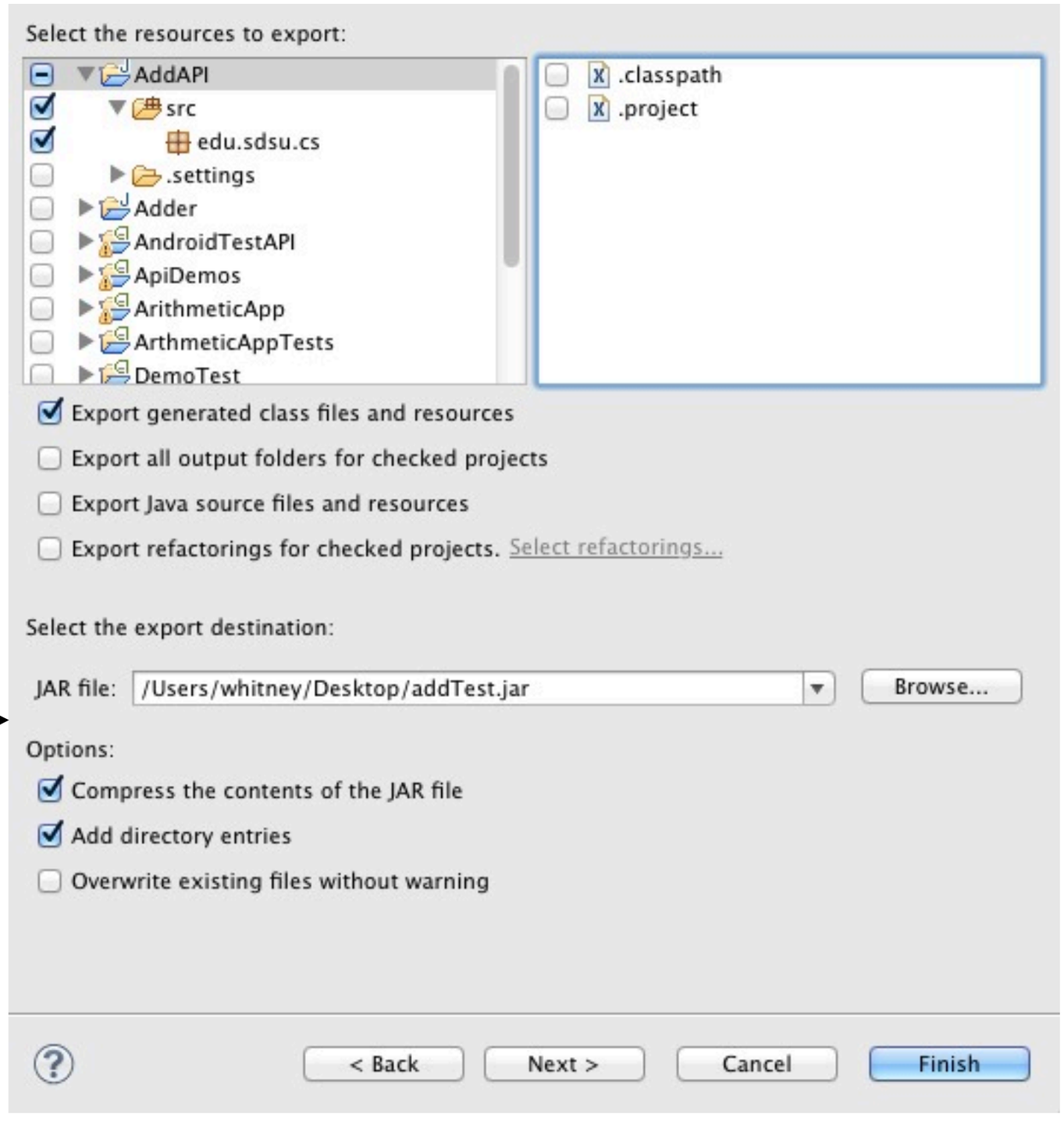
Java Class in Java project

```
package edu.sdsu.cs;

public class Add {
	public int add(int a, int b) {
		return a + b;
	}
}
```



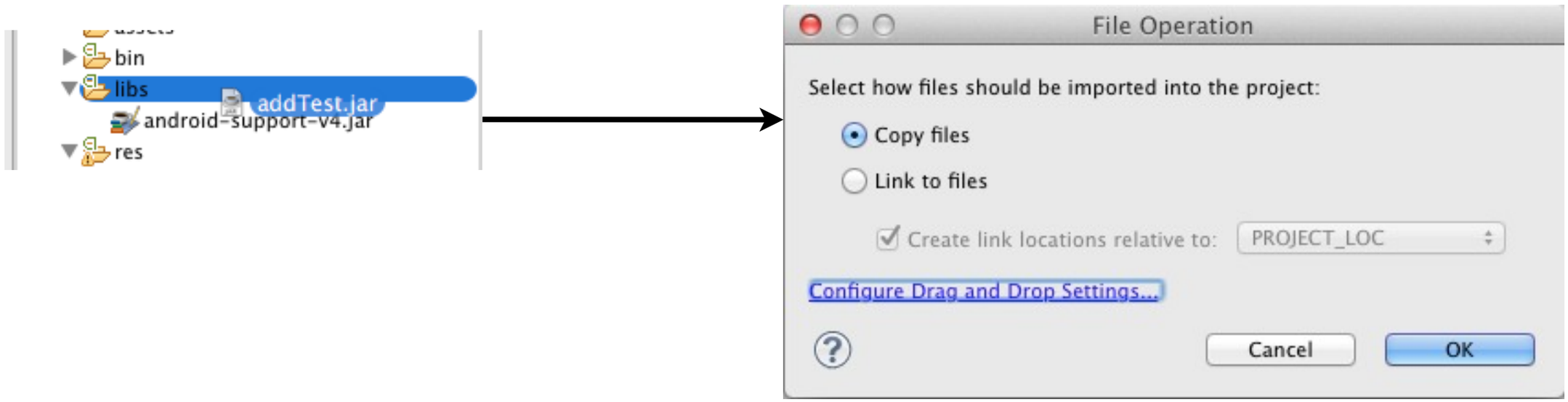
Exporting Java Jar File

Export Java Project



Importing Java jar file into Android Project

Drag and drop jar file into libs directory of Android Project



Make sure files are copied into project for this class

Using the Add Class in Android Project

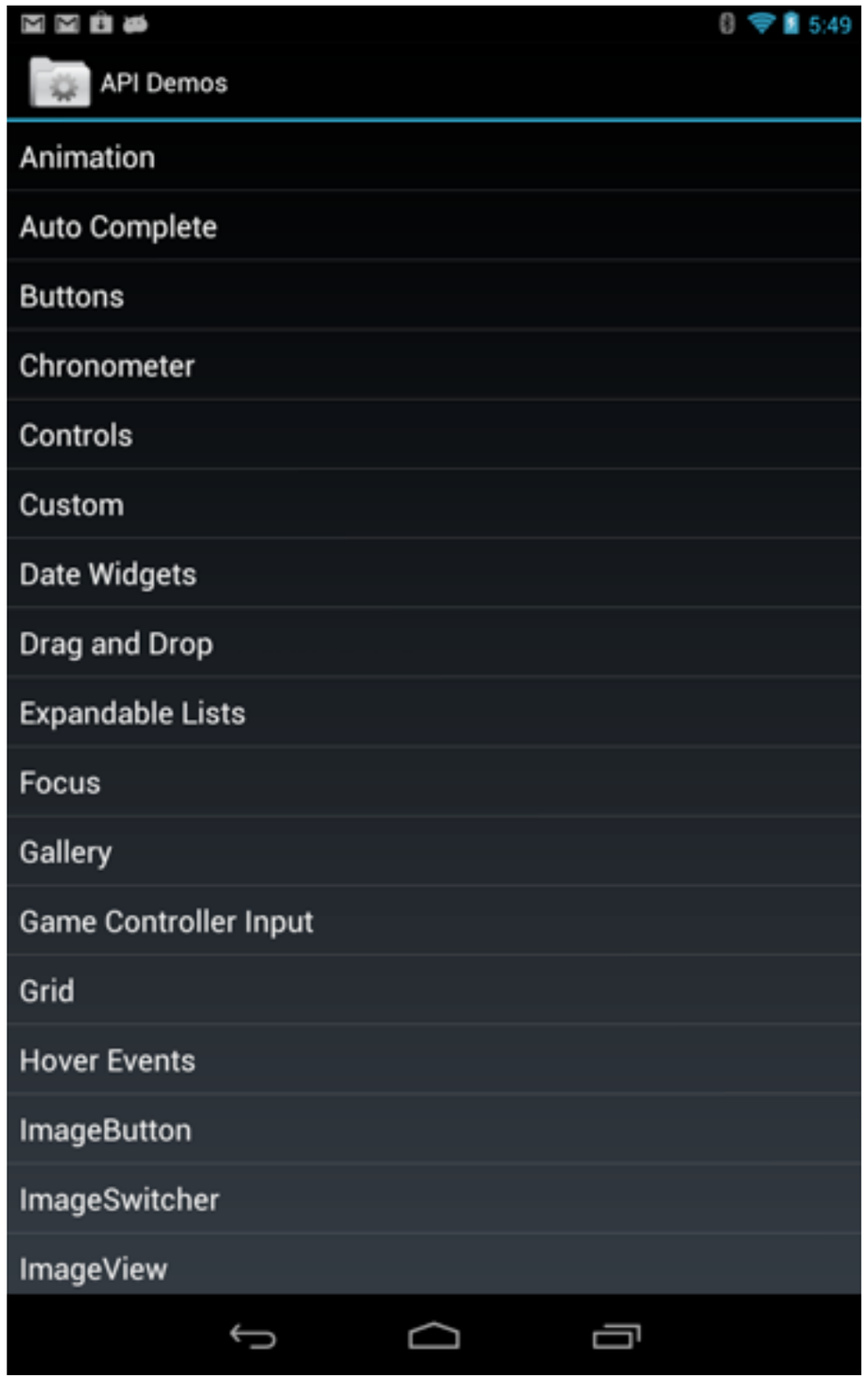
```
import edu.sdsu.cs.Add;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.TextView;

public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
            Add adder = new Add();
            int result = adder.add(1,2);
            TextView answer = (TextView) findViewById(R.id.answer);
            answer.setText("" + result);
    }
```

Android Examples

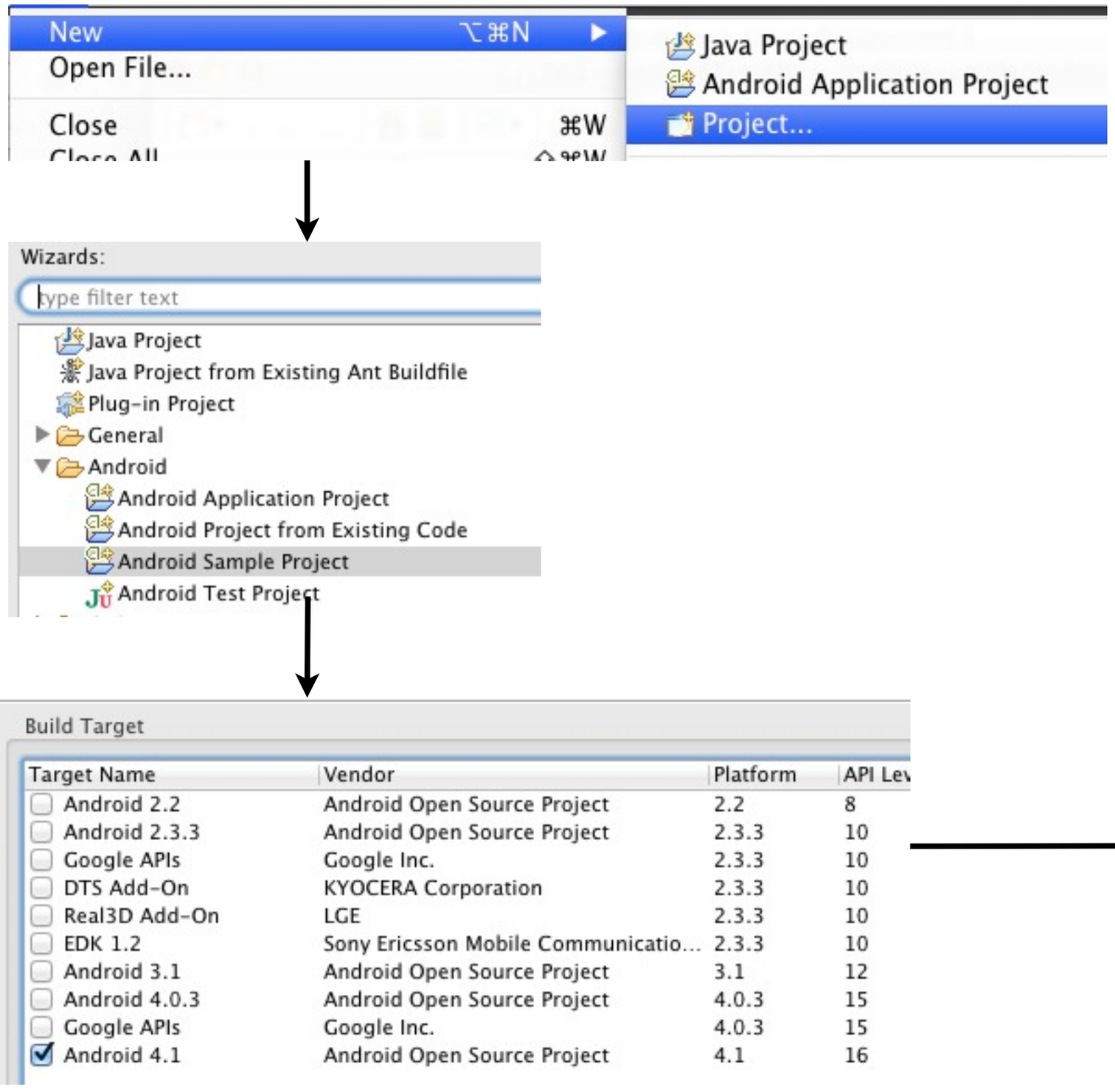
API Demos

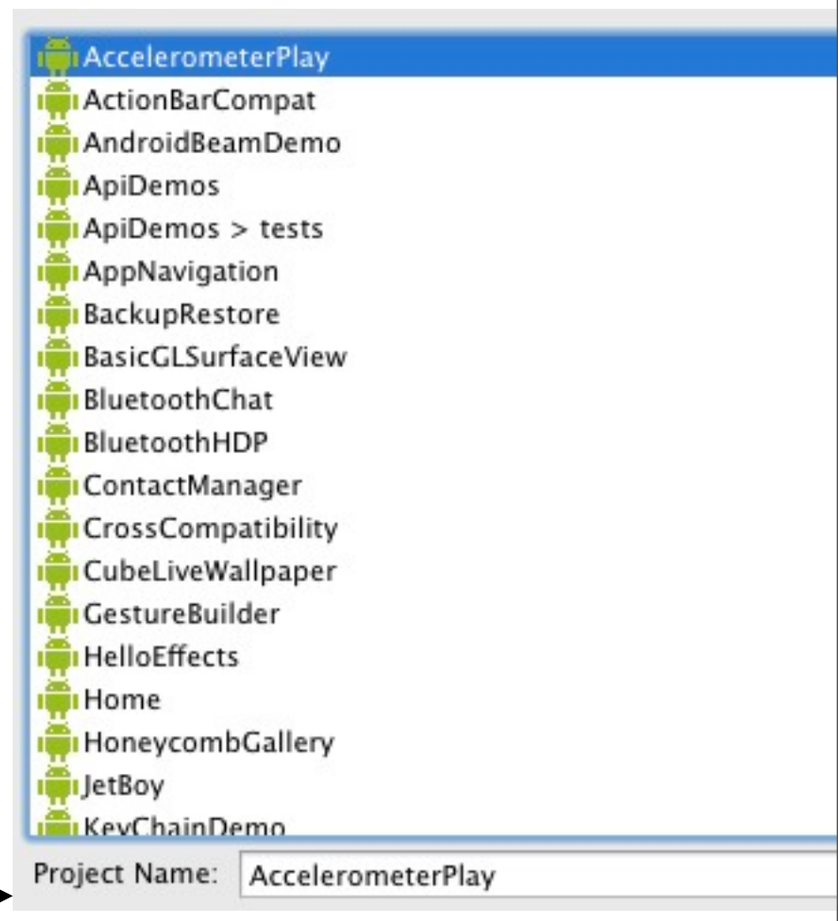
Show many features of Android UI

Source code



Loading Android Examples





Lists

Lists

ListActivity handled much of work

List View

Adapter for the data

Lists - Example

```
public class SelectionExamples extends ListActivity {
        String[] items = { "Gautama Buddha", "Kalki", "Krishna",
                        "Kurma", "Matsya", "Narasimha", "Parashurama",
                        "Rama", "Vamana", "Varaha" };


        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setListAdapter(new ArrayAdapter<String>(this,
                                android.R.layout.simple_list_item_1, items));
        }
}
```

Layout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
 <TextView
    android:id="@+id/selection"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" android:text="@string/list_title"/>
  <ListView
    android:id="@android:id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:drawSelectorOnTop="false"
    />
</LinearLayout>
```


If the list view has id "@android:id/list" the ListActivity will find it without having to call setContentView

Adapter Pattern

ArrayAdapter

```
setListAdapter(new ArrayAdapter<String>(this,
                        android.R.layout.simple_list_item_1, items));
```

We have an array of Strings

ListView uses a list of Views

ArrayAdapter

- Adapts (converts) what we have to what ListView needs
- Creates a view for each element in the array

Other Adapters

CursorAdapter

Converts database cursor for display in selection view

SimpleAdapter

Converts XML resources

ArrayAdapter

Can

- Add
- Remove
- Filter
- Sort

public ArrayAdapter (Context context, int textViewResourceId, T[] objects)

public ArrayAdapter (Context context, int resource, int textViewResourceId, T[] objects)

Supply view to populate

Supplied Text views

android.R.layout.simple_list_item_1

android.R.layout.simple_list_item_2

two rows, not compatable with array adapter

android.R.layout. two_line_list_item

android.R.layout.simple_list_item_activated_1

ADK level 11 or greater

android.R.layout.simple_list_item_activated_2

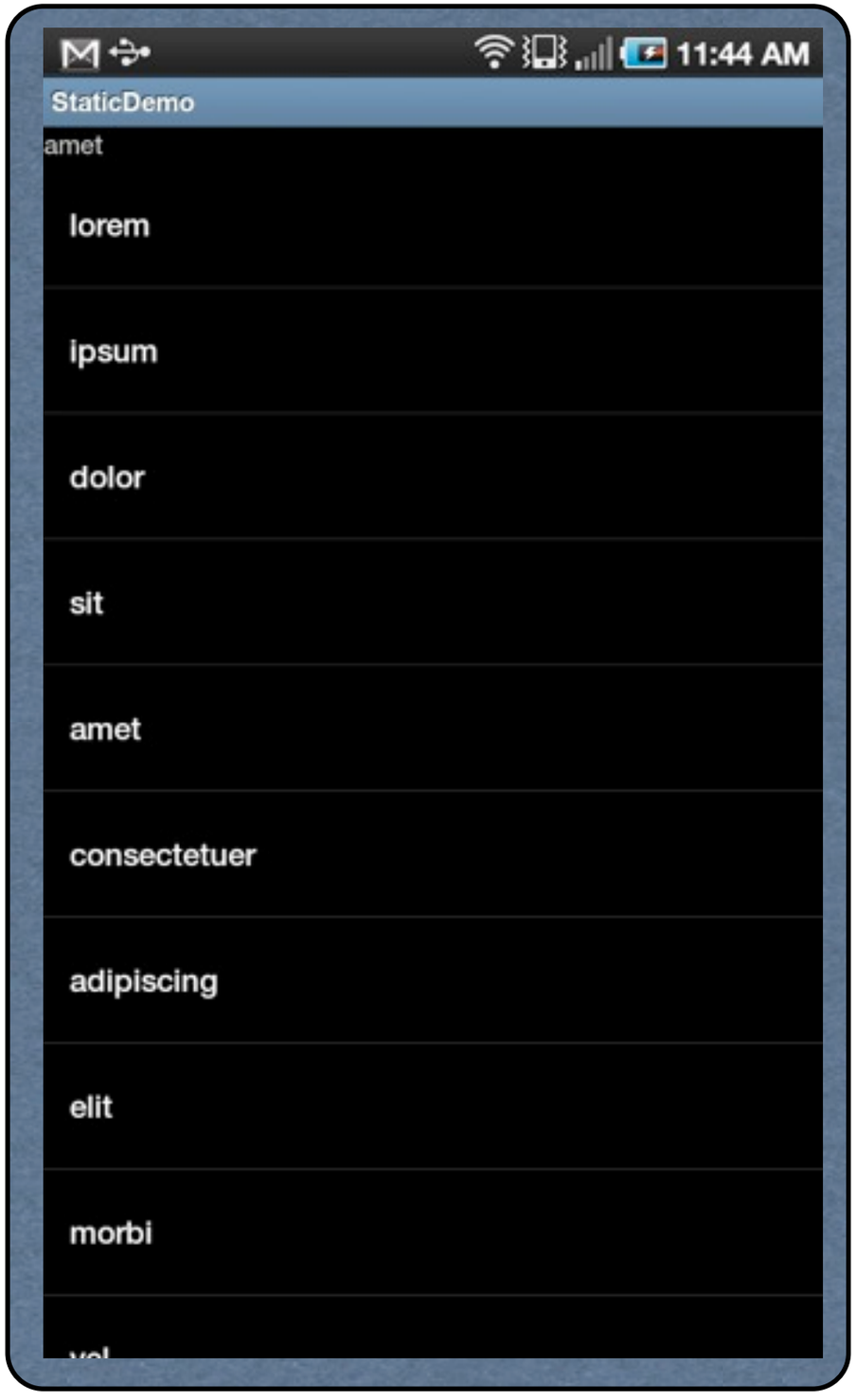
ADK level 11 or greater

android.R.layout.simple_list_item_single_choice

android.R.layout.simple_list_item_multiple_choice

Supplied Text views

android.R.layout.simple_list_item_1

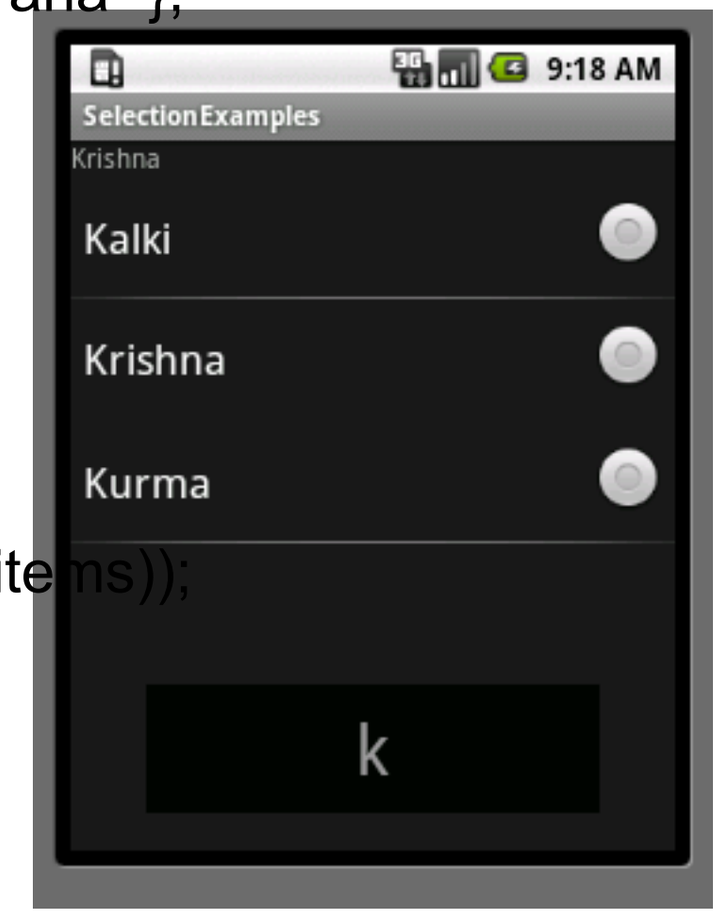


Supplied Text views

android.R.layout.simple_list_item_1

Single Choice with Text Filter

```
public class SelectionExamples extends ListActivity {
    String[] items = { "Gautama Buddha", "Kalki", "Krishna", "Kurma", "Matsya",
            "Narasimha", "Parashurama", "Rama", "Vamana", "Varaha" };
    TextView selection;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        setListAdapter(new ArrayAdapter<String>(this,
                android.R.layout.simple_list_item_single_choice, items));
        selection = (TextView) findViewById(R.id.selection);
        getListView().setTextFilterEnabled(true);
    }

    public void onListItemClick(ListView parent, View v, int position, long id) {
        selection.setText((String)getListView().getItemAtPosition(position));
    }
}
```

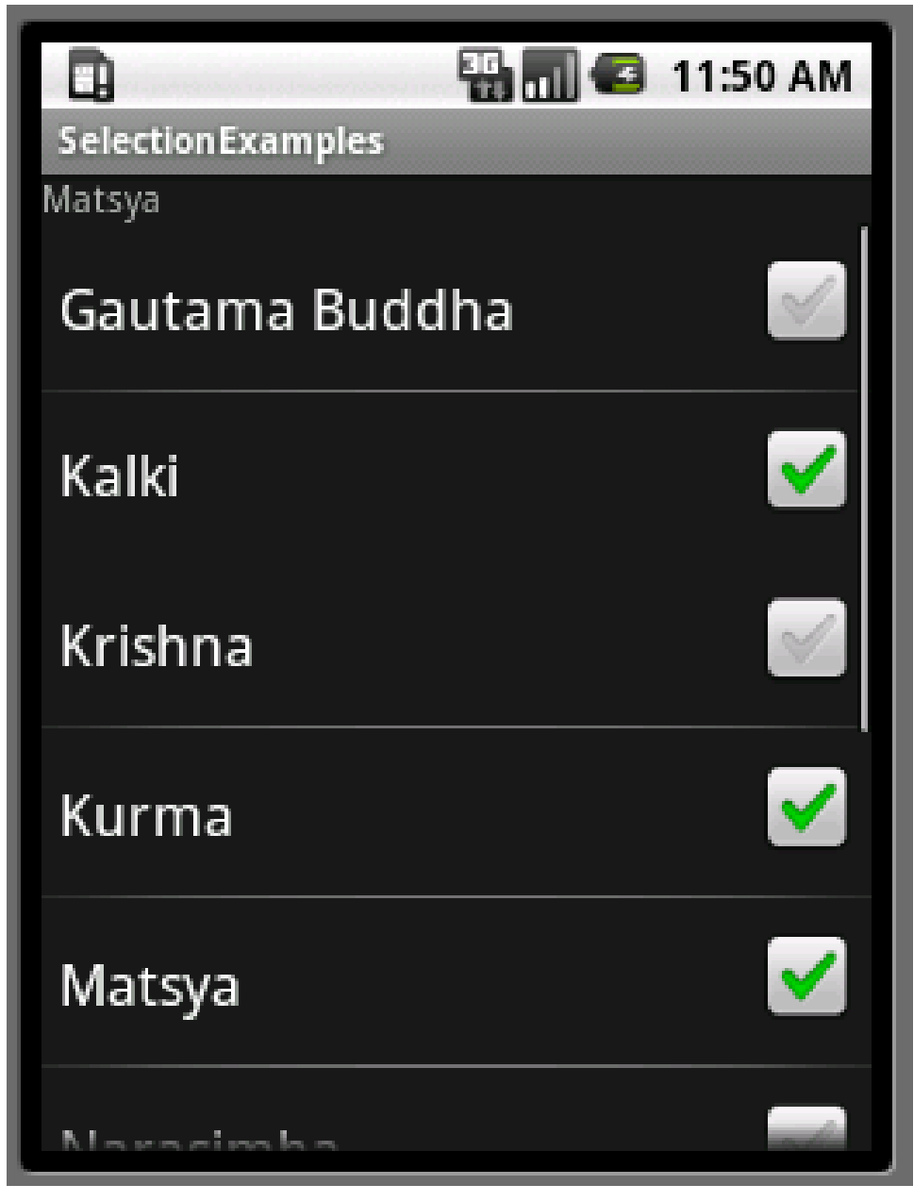


MultiSelection

```
public class SelectionExamples extends ListActivity {
    String[] items = { "Gautama Buddha", "Kalki", "Krishna", "Kurma", "Matsya",
            "Narasimha", "Parashurama", "Rama", "Vamana", "Varaha" };
    TextView selection;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        setListAdapter(new ArrayAdapter<String>(this,
                android.R.layout.simple_list_item_multiple_choice, items));
        selection = (TextView) findViewById(R.id.selection);
        getListView().setTextFilterEnabled(true);
        getListView().setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
    }

    public void onListItemClick(ListView parent, View v, int position, long id) {
        selection.setText((String)getListView().getItemAtPosition(position));
    }
}
```


When you want to find out all the items that were selected you need to call getCheckedItemPositions() on the list view.

MultiSelection



Using Resources

Can use list from resources

```
public class SelectionExamples extends ListActivity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
                this, R.array.avatars, android.R.layout.simple_list_item_1);
        setListAdapter(adapter);
    }
}
```

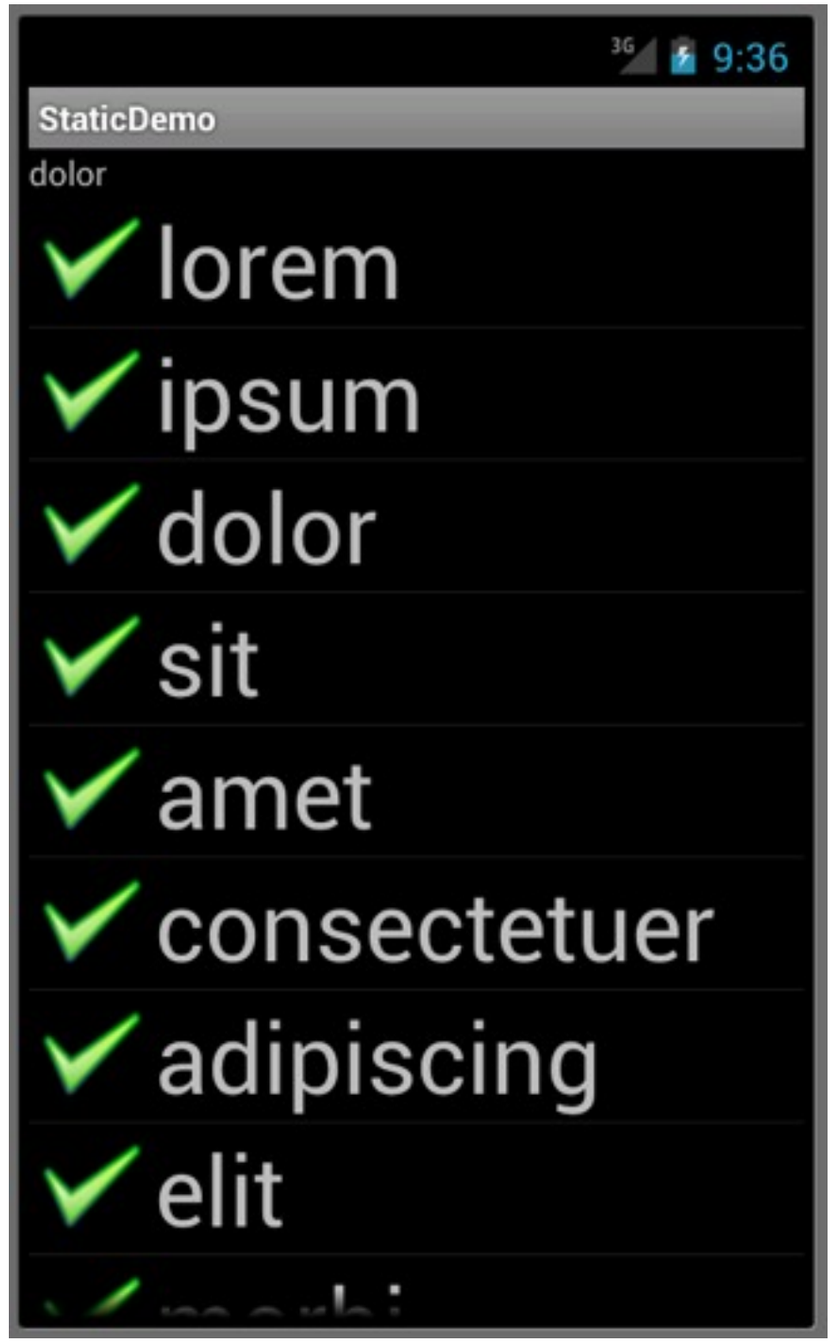
Strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="list_title">Ten avatars</string>
    <string name="app_name">Selection Examples</string>
      <string-array name="avatars">
            <item>Gautama Buddha</item>
            <item>Kalki</item>
            <item>Krishna</item>
            <item>Kurma</item>
            <item>Matsya</item>
            <item>Narasimha</item>
            <item>Parashurama</item>
            <item>Rama</item>
            <item>Vamana</item>
            <item>Varaha</item>
      </string-array>
</resources>
```

Using your Own Row Layout

res/layout/row.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
              android:id="@+id/icon"
              android:padding="2dip"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:src="@drawable/ok"
        />
        <TextView
              android:id="@+id/label"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:textSize="40sp"
        />
</LinearLayout>
```



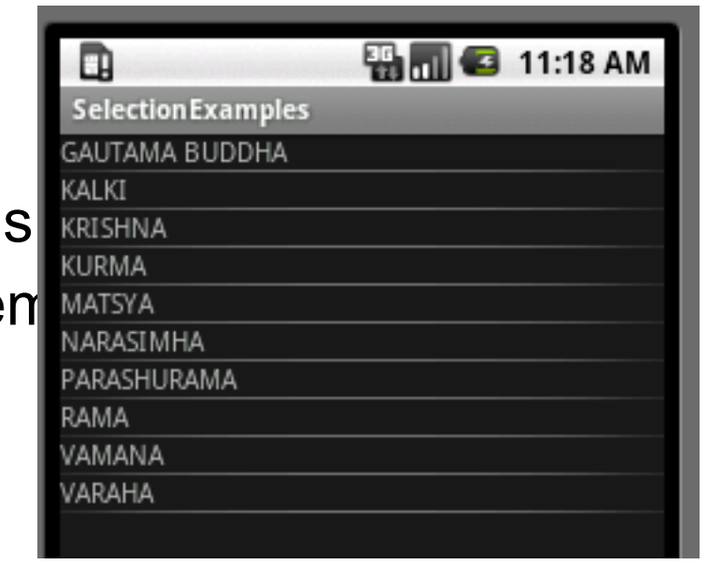
Activity

```
public class StaticDemo extends ListActivity {
    private TextView selection;
    private static final String[] items={"lorem", "ipsum", "dolor",
                        "sit", "amet","consectetuer", "adipiscing", "elit", "morbi", "vel",
                        "ligula", "vitae", "arcu", "aliquet", "mollis", "etiam", "vel", "erat", "placerat", "ante"};

    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);
        setContentView(R.layout.main);
        setListAdapter(new ArrayAdapter<String>(this,
                                                R.layout.row, R.id.label,
                                                items));
        selection=(TextView)findViewById(R.id.selection);
    }

    public void onListItemClick(ListView parent, View v, int position, long id) {
        selection.setText(items[position]);
    }
}
```


Example from Chapter 13 of Beginning Android 4. Note the ArrayAdapter constructor we give the layout and the reference to the text field to fill out.

Creating an Adapter

```
public class SelectionExamples extends ListActivity {
    String[] items = { "Gautama Buddha", "Kalki", "Krishna", "Kurma", "Matsya",
            "Narasimha", "Parashurama", "Rama", "Vamana", "Varaha" };

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new UpperCaseArrayAdapter(this
                android.R.layout.simple_list_item_1, item
    }
}
```



UpperCaseArrayAdapter

```
public class UpperCaseArrayAdapter extends ArrayAdapter<String> {
        private Context mContext;
        private String[] mElements;
        public UpperCaseArrayAdapter(Context context, int resource,
                        String[] elements) {
                super(context, resource, elements);
                mContext = context;
                mElements = elements;
        }

        public View getView(int position, View convertView, ViewGroup parent) {
                if (convertView == null) {
                        convertView = new TextView(mContext);
                }
                ((TextView) convertView).setText(mElements[position].toUpperCase());
                return (convertView);
        }
}
```

LayoutInflater

Reads a layout file and returns the UI objects

So rather than create UI elements in code

Can read file and inflate when needed

LayoutInflator example

In UpperCaseArrayAdapter we create a new TextView

```
public View getView(int position, View convertView, ViewGroup parent) {
        if (convertView == null) {
            convertView = new TextView(mContext);
        }
        ((TextView) convertView).setText(mElements[position].toUpperCase());
        return (convertView);
    }
```

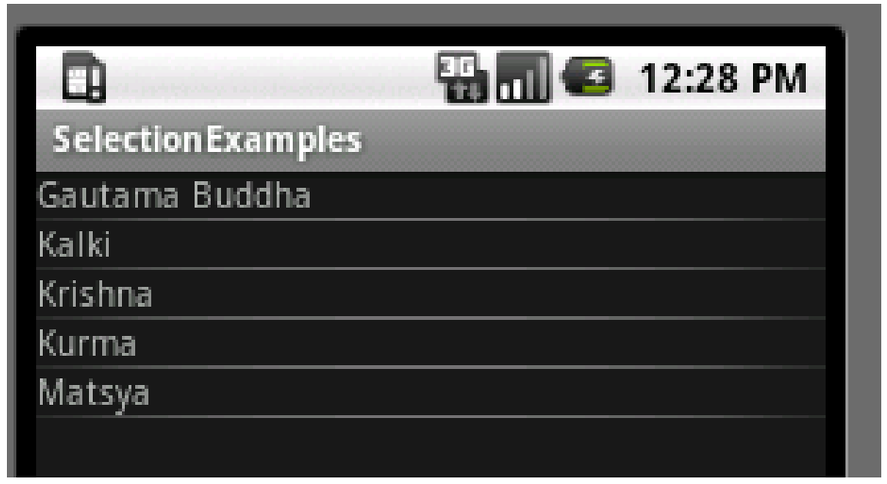
But what if we want a fancy row with multiple items

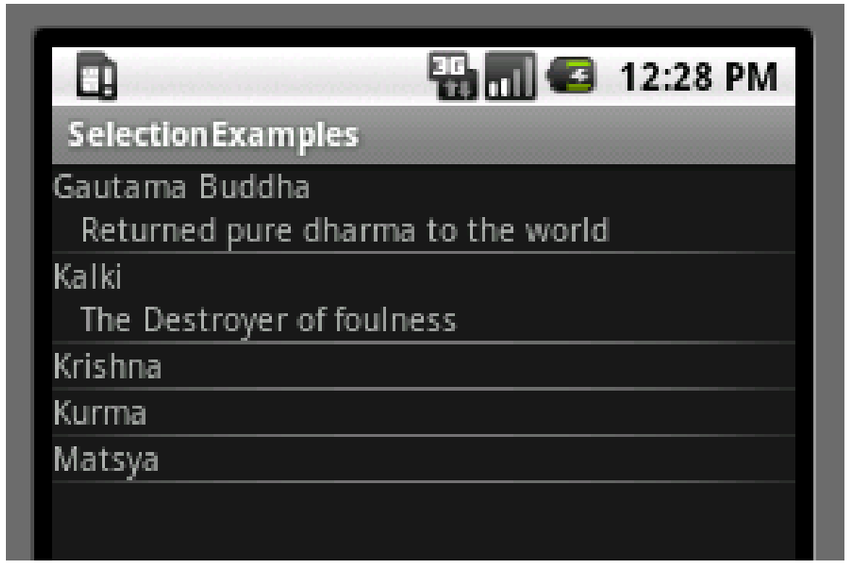
LayoutInflator example

```
public View getView(int position, View convertView, ViewGroup parent) {
        if (convertView == null) {
                LayoutInflater inflater=getLayoutInflater();
            convertView =inflater.inflate(R.layout.row, parent, false);
        }
    TextView textForRow=(TextView)row.findViewById(R.id.rowText);

        textForRow.setText(mElements[position].toUpperCase());
        return (convertView);
}
```


R.layout.row - the layout to read,
parent - the view to get parent view attributes and attach the new view
false - false meaning do not attach the new view to the parent. In this case we return the new view and it gets attached for us

Expandable List





The Activity

```
public class SelectionExamples extends ListActivity {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new AvatarListAdapter(this));
    }

    protected void onListItemClick(ListView l, View v, int position, long id) {
        ((AvatarListAdapter) getListAdapter()).toggle(position);
    }
```

AvatarListAdapter Data

```
private class AvatarListAdapter extends BaseAdapter {
        private Context mContext;

        private String[] mAvatars = { "Gautama Buddha", "Kalki", "Krishna",
                "Kurma", "Matsya" };

        private String[] mAvatarDescription = {
                "   Returned pure dharma to the world",
                "   The Destroyer of foulness",
                "   Represents a person in more practical society",
                "   The tortoise, represents a human embryo just growing tiny legs, with a hug
                "   The fish, appeared in the Satya Yuga. Represents beginning of life" };

    private boolean[] mExpanded = { false, false, false, false, false };
```

AvatarListAdapter Accessors

```
public AvatarListAdapter(Context context) {
        mContext = context;
}

public int getCount() {
        return mAvatars.length;
}

public Object getItem(int position) {
        return position;
}

public long getItemId(int position) {
        return position;
}
```

AvatarListAdapter getView

```
public View getView(int position, View convertView, ViewGroup parent) {
    AvatarView avatar;
    if (convertView == null) {
        avatar = new AvatarView(mContext, mAvatars[position],
                mAvatarDescription[position], mExpanded[position]);
    } else {
        avatar = (AvatarView) convertView;
        avatar.setTitle(mAvatars[position]);
        avatar.setDialogue(mAvatarDescription[position]);
        avatar.setExpanded(mExpanded[position]);
    }
    return avatar;
}

public void toggle(int position) {
    mExpanded[position] = !mExpanded[position];
    notifyDataSetChanged();
}
}
```

AvatarView

```
private class AvatarView extends LinearLayout {

    private TextView mTitle;
    private TextView mDescription;

    public void setTitle(String title) {
        mTitle.setText(title);
    }

    public void setDialogue(String words) {
        mDescription.setText(words);
    }

    public void setExpanded(boolean expanded) {
        mDescription.setVisibility(expanded ? VISIBLE : GONE);
    }
```

AvatarView Constructor

```
public AvatarView(Context context, String title, String dialogue,
        boolean expanded) {
    super(context);

    this.setOrientation(VERTICAL);

    mTitle = new TextView(context);
    mTitle.setText(title);
    addView(mTitle, new LinearLayout.LayoutParams(
            LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT));

        mDescription = new TextView(context);
        mDescription.setText(dialogue);
        addView(mDescription, new LinearLayout.LayoutParams(
                LayoutParams.FILL_PARENT,
LayoutParams.WRAP_CONTENT));

        mDescription.setVisibility(expanded ? VISIBLE : GONE);
    }
```