CS 696 Mobile Phone Application Development
Fall Semester, 2009
Doc 8 Lists
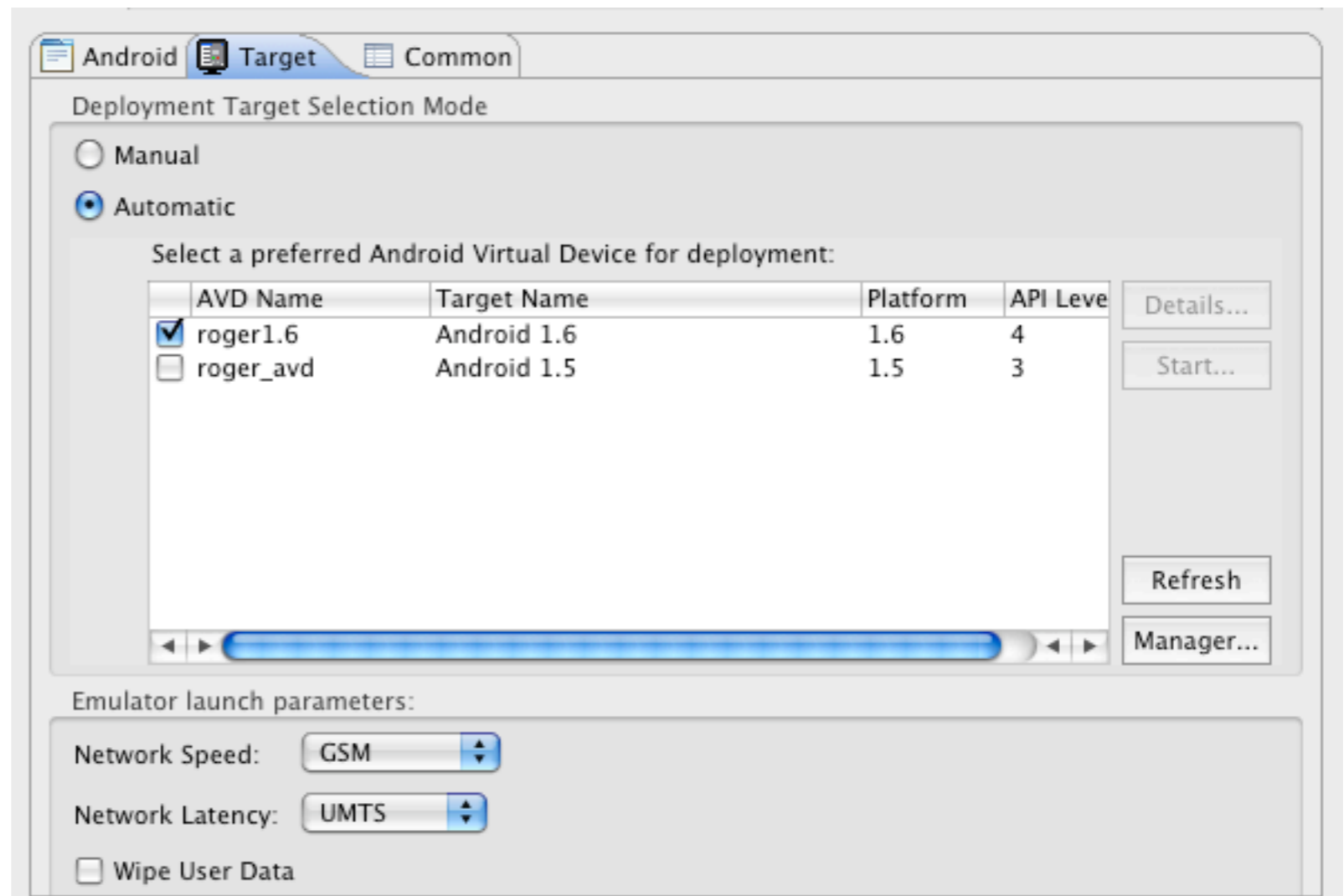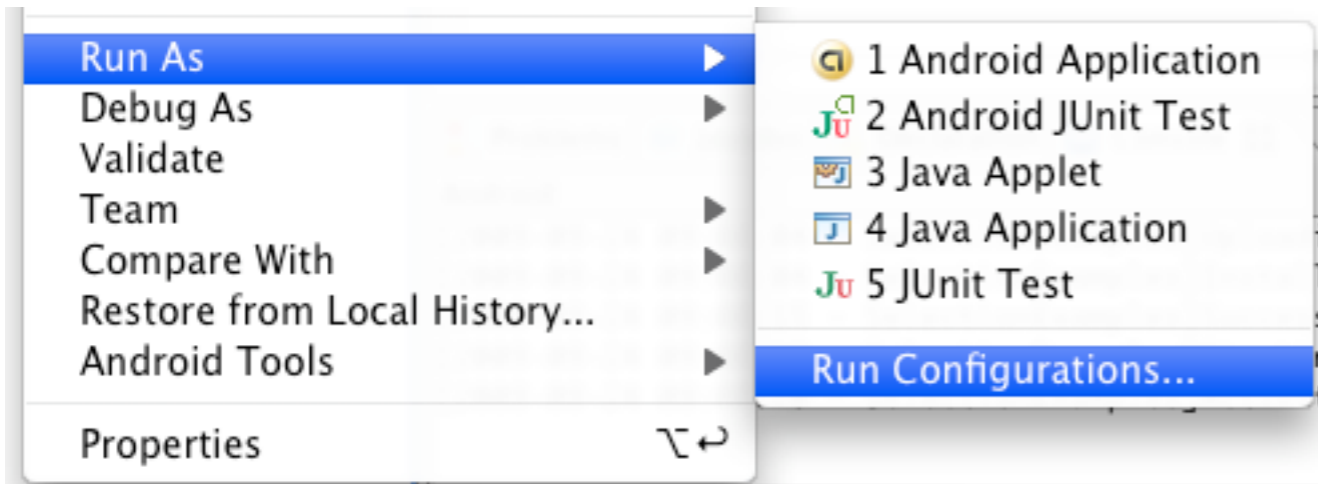## Sept 24, 2009

References

The Busy Coder's Guide to Android Development, V2.1, Mark L. Murphy
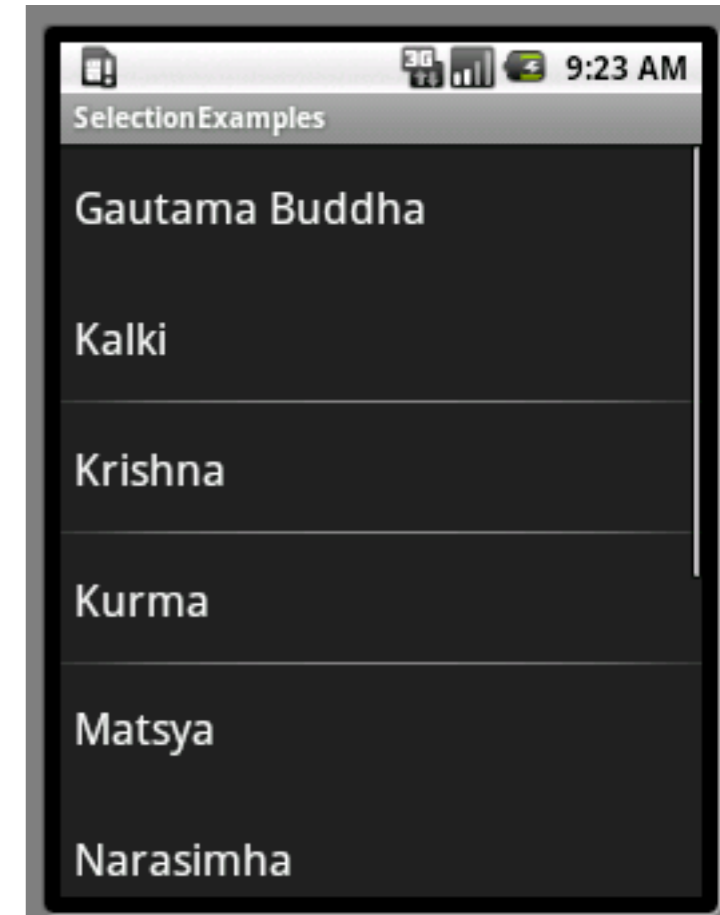
Android List Examples

Run Configurations

Lists

```
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
<?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));
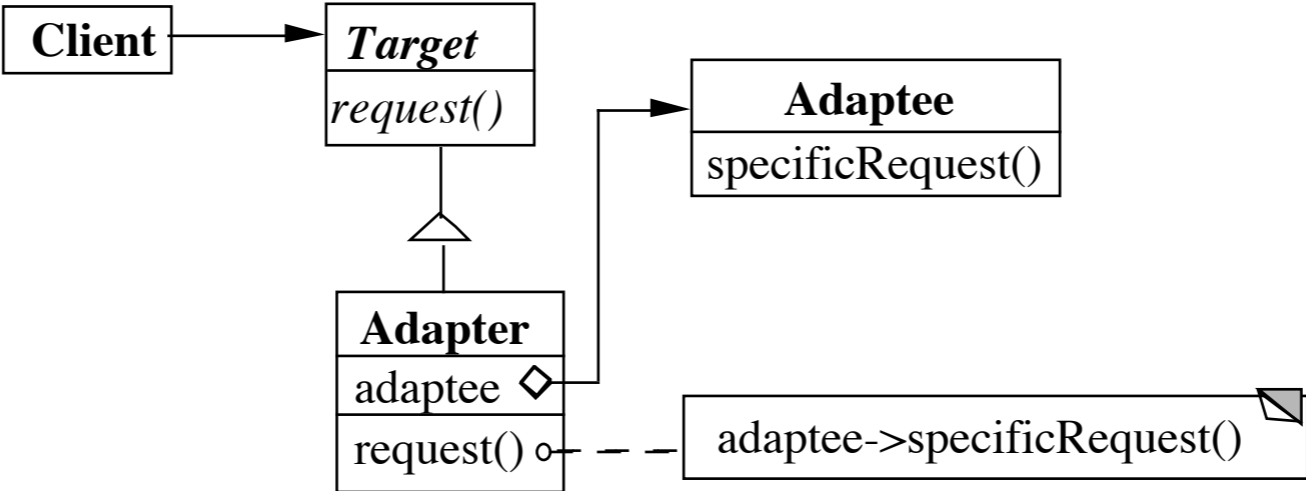
We have an array of Strings

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

Object Adapter

Client → **Target** *request( )*

Adaptee specificRequest()

Adapter adaptee ◇ request() ○ - - - adaptee->specificRequest()

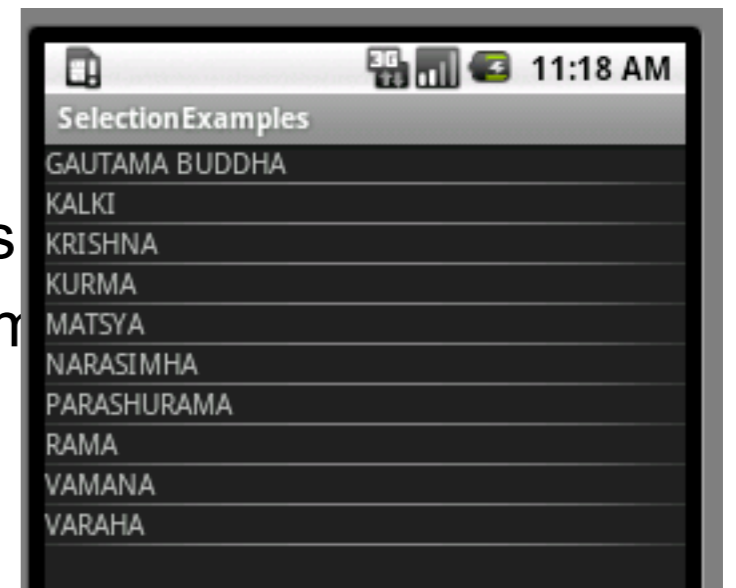# Other Adapters

CursorAdapter

    Converts database cursor for display in selection view

SimpleAdapter

    Converts XML resources

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

```java
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);
    }
}
```

Using Resources

```java
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
<?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>
```
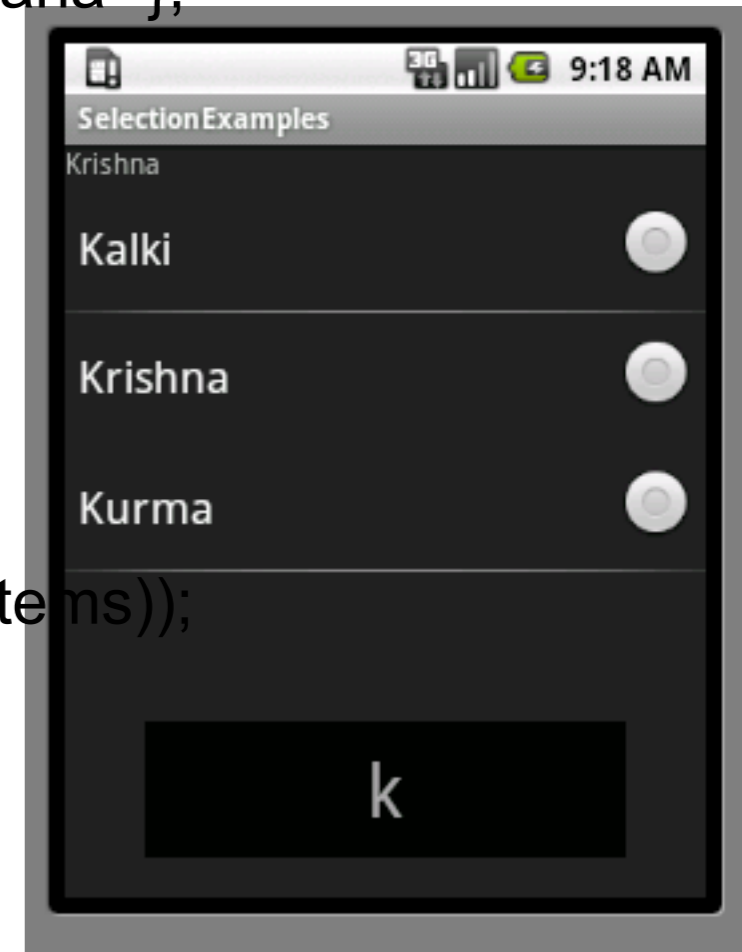
Single Choice with Text Filter

```java
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

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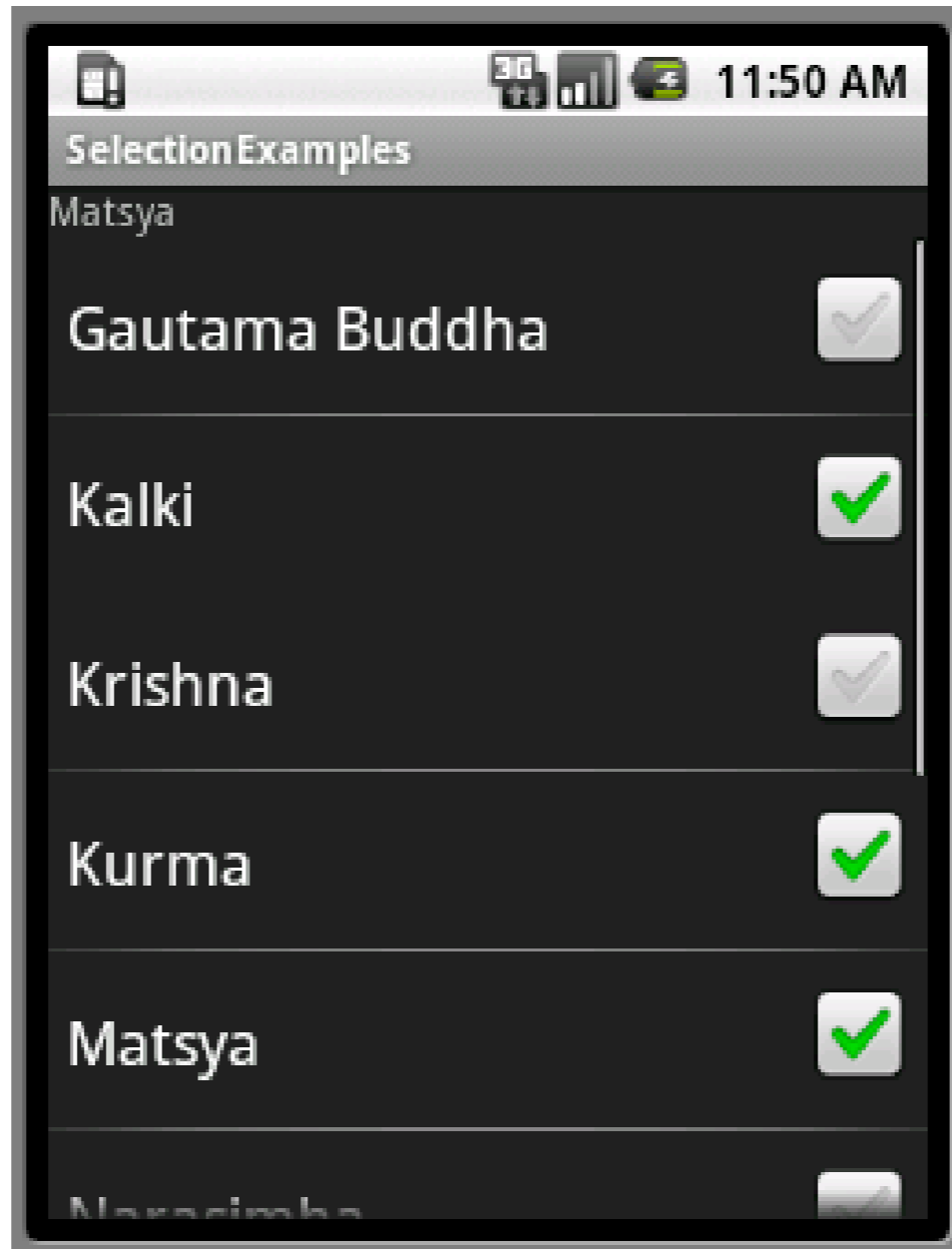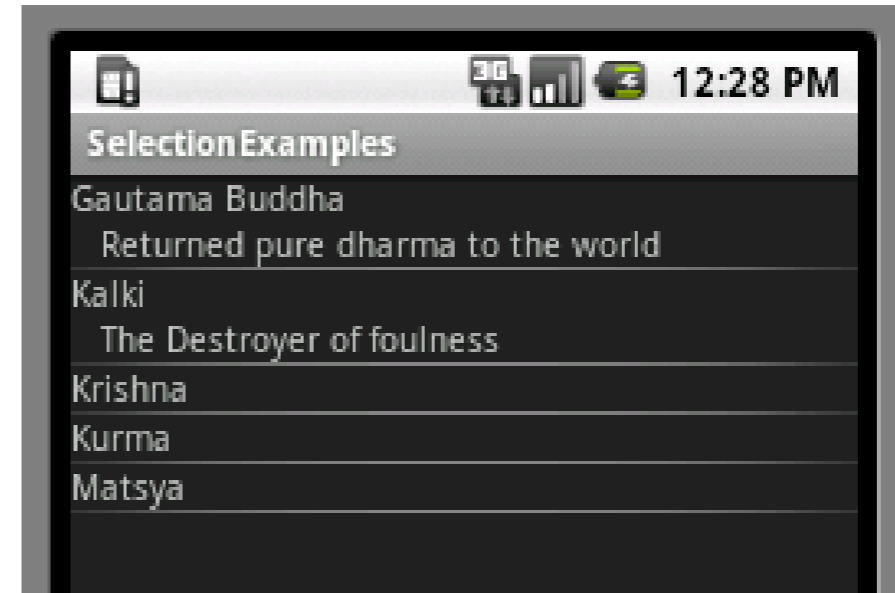
When you want to find out all the items that were selected you need to call getCheckedItemPositions() on the list view.

MultiSelection

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

```java
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

```java
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

```java
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

```java
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);
}
```