CS 646 Android Mobile Application Development
Spring Semester, 2015
Doc 6 SDK Levels, Fragments
Feb 5, 2015

Min & Target SDK

Min and Target SDK


```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"
    />
```

Min and Target SDK - Eclipse

Stored in Manifest file

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="edu.sdsu.cs696"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".IntentExample"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="2" />
</manifest>
```

Min and Target SDK - Android Studio

Stored in projectDirectory/app/build.gradle

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "edu.sdsu.cs.whitney.testapilevel"
        minSdkVersion 17
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
```

minSdkVersion

Earliest version that your app works in

Default value 8

Always set this

Market will not show app to devices with lower SDK

Device will not load app if device has lower SDK

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"
    />
```

targetSdkVersion

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"
    />
```

You have made sure that app runs on
SDK from minSdkVersion and targerSdkVersion

Default value minSdkVersion

Build SDK Version

Which Java class files to use when compiling

Use latest version

Beyond Target SDK

What happens is target SDK is 16 and run on device with SDK 17?

Android compatibility mode

Android "turns off" features not supported in target SDK

Between min SDK and Target SDK

Or why have both?

App may use features of target SDK that are not required for operation of app

Example: gestures, but have menu items for gesture

So app can run on lower SDK that target

Compiler/lint creates error if call method added to Android after min SDK

How to Support "new" methods

Example

finishAffinity () - added in SDK 16

finish() - added in SDK 1

In Activity class

```
private void quit() {
    finishAffinity();
}
```

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"
    />
```

Compiler error in Eclipse

Android Studio - underlined

How to Support "new" methods

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"
    />
```

```
@TargetApi(Build.VERSION_CODES.JELLY_BEAN)
private void quit() {
    finishAffinity();
}
```

No compiler error

Run time error on non-SDK16 devices

How to Support "new" methods

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"
    />
```

```
@TargetApi(Build.VERSION_CODES.JELLY_BEAN)
private void quit() {
    int apiVersion = android.os.Build.VERSION.SDK_INT;
    if (apiVersion >= android.os.Build.VERSION_CODES.JELLY_BEAN){
            finishAffinity();
    } else{
            finish();
        }
}
```

No compiler error

No runtime errors

How to Support "new" methods

```
<uses-sdk
    android:minSdkVersion="8"
    android:targetSdkVersion="16"
    />
```

```
@TargetApi(16)
private void quit() {
    int apiVersion = android.os.Build.VERSION.SDK_INT;
    if (apiVersion >= 16){
            finishAffinity();
    } else{
            finish();
        }
}
```

You can use the version number directly

@TargetApi verses @SuppressLint("NewApi")

@TargetApi(16)

Turns off warnings for methods/classes added

After min SDK and in SDK 16 or earlier

Will get warnings for using features after

min SDK and SDK 16

@SuppressLint("NewApi")

Turns off all warnings for using any feature added after min SDK

Avoid using this

Can introduce errors later

Inflation

Inflation

Converting XML file into Java objects

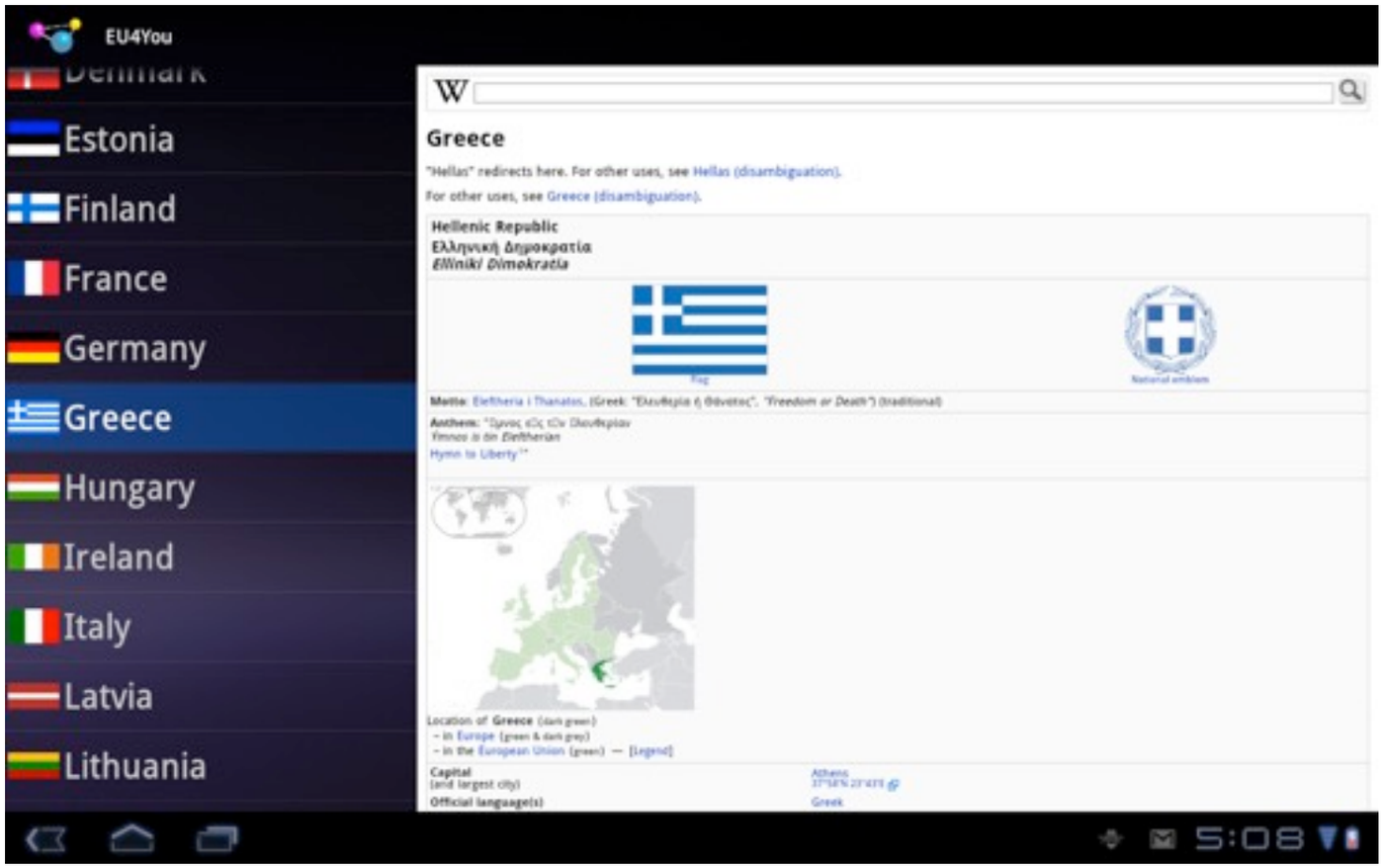
Android Activity Project

```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }

}
```

Fragments

Motivation

Display in one screen on tablet



2 activities on Phone

Fragments

Like a sub-activity

Has

- Java class (subclass of Fragment)
- Layout
- Life cycle methods

Embed in an activity

Fragments & Android Versions

Added in Android 3.0

Back ported to Android 1.6

- Need to add Android Compatibility Library
- Few differences with Fragments in Android 3.0+

Fragment Issues

Creating fragments

Adding layout to fragment

Connecting fragment to activity

Activity talking to fragment

Fragment talking to Activity

Fragment & Activity lifecycle

Using Fragments in Android 2

Fragments and back stack

Creating Fragment

Subclass android.app.Fragment

New life cycle methods added

onCreateView

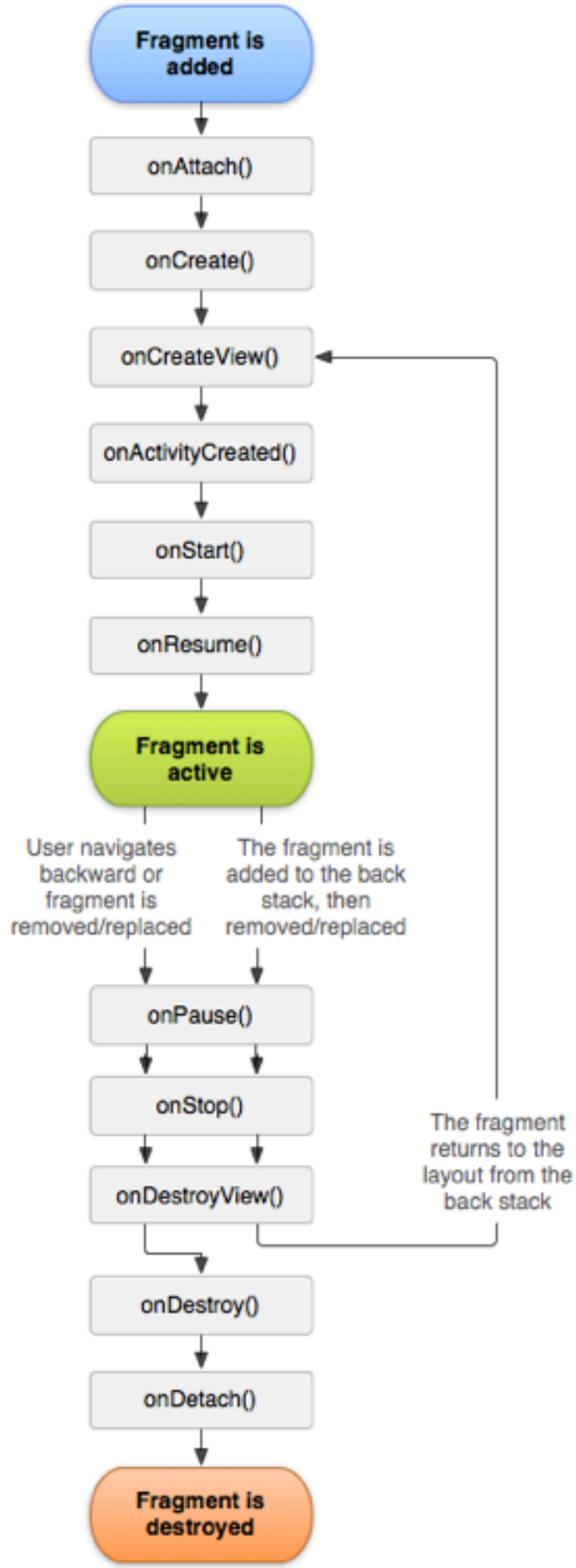
Create the fragment view layout or in code

onAttach

When fragment is associated with activity

onActivityCreated

When activity is created



Fragment Limitation

Does not support

onRetainNonConfigurationInstance()

Special Fragment subclasses

DialogFragment

ListFragment

Has special mode to keep selected row highlighted

PreferenceFragment

Adding a View to a Fragment - Layout

```
public class ResultFragment extends Fragment {
    private TextView inputText;

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState) {
        View resultView =  inflater.inflate(R.layout.result_fragment, container, false);
        inputText = (TextView) resultView.findViewById(R.id.result_text);
        return resultView;
    }
```

Fragment Layout - Just normal layout

res/layout/result_fragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <TextView
        android:id="@+id/result_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="You typed: "
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <TextView
        android:id="@+id/result_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

Adding Fragment to Activity - Layout

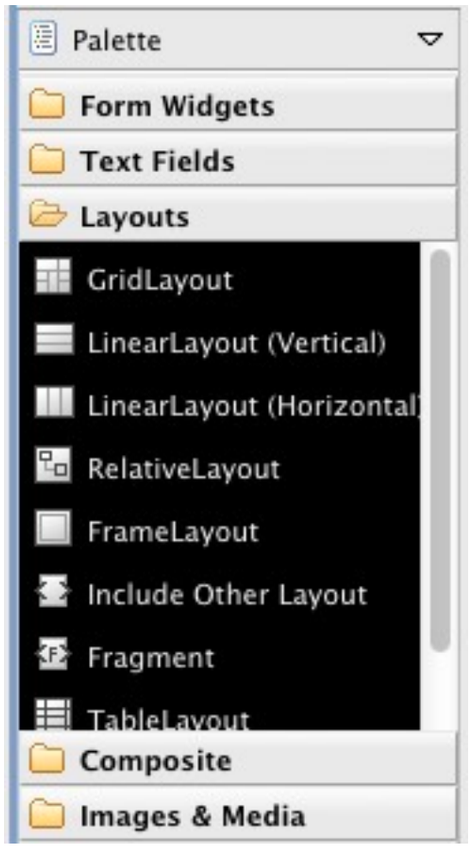
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView14"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:text="Welcome to the Fragment world"
        android:textAppearance="?android:attr/textAppearanceLarge" />
 <fragment android:name="edu.sdsu.cs.whitney.EditFragment"
            android:id="@+id/edit_fragement"
            android:layout_weight="5"
            android:layout_width="match_parent"
            android:layout_height="0dp" />
<fragment android:name="edu.sdsu.cs.whitney.ResultFragment"
            android:id="@+id/result_fragment"
            android:layout_weight="5"
            android:layout_width="match_parent"
            android:layout_height="0dp" />
</LinearLayout>
```

Fragment tag Details

<fragment android:name="fullClassNameOfFragment"

<fragment android:name="edu.sdsu.cs.whitney.EditFragment"
android:id="@+id/edit_fragment"

ID needed for accessing fragment



Accessing Fragment in Activity

```
FragmentManager fragments = getFragmentManager();
ResultFragment resultFragment =
        (ResultFragment)fragments.findFragmentById(R.id.result_fragment);
```

Accessing Fragment in Activity - Compatibility

Activity class

```
FragmentManager manager = getFragmentManager();
```

Compatibility classes (ActionBarActivity class)

```
FragmentManager manager = getSupportFragmentManager();
```

Accessing Activity in Fragment

getActivity()

returns activity that contains fragment

public void onAttach(Activity activity)

Method called on fragment

Argument is activity that contains fragment

Adding Fragment to Activity - In Activity

```
public class FragmentsSimpleExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
            FragmentManager fragments = getFragmentManager();
            FragmentTransaction fragmentTransaction =
fragments.beginTransaction();
            EditFragment fragment = new EditFragment();
            fragmentTransaction.add(R.id.fragment_holder, fragment);
            fragmentTransaction.commit();
        }
}
```

Need view to add the fragment into

main.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="fill_parent"
    android:orientation="vertical" >

     <FrameLayout
        android:id="@+id/fragment_holder"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </FrameLayout>

</LinearLayout>
```

Dynamically Replacing Fragments

Can replace fragments dynamically

So can change part of view at runtime

Can make change part of the back stack

if add transaction to back stack

Back button brings back previous fragment

```
FragmentManager fragments = getFragmentManager();
FragmentTransaction fragmentTransaction = fragments.beginTransaction();
ResultFragment fragment = new ResultFragment();
fragmentTransaction.replace(R.id.fragment_holder, fragment);
fragmentTransaction.addToBackStack("Optional Name");
fragmentTransaction.commit();
```

Design Issues

Fragment isolation

ID clashes

Fragment isolation

If fragments can be used in different activities

Fragment logic must be self contained

Fragment talks to containing activity without knowing which activity

Fragment isolation - Solution

Fragment defines listener interface

Activity implements fragment listener interface

Fragment sends data/events to activity using interface

Fragments, Activities, Dependence

- Fragments are reusable
  - Can not know about activity
    - Name of activity
    - Methods in activity
    - Activity fields
- Activities are not reusable
  - Can know about fragments
    - Class name
    - methods

ID clash

Fragment and Activity widget ids
Are stored in same id class in R file

How make sure that the id names are not the same

In example use fragment name in id

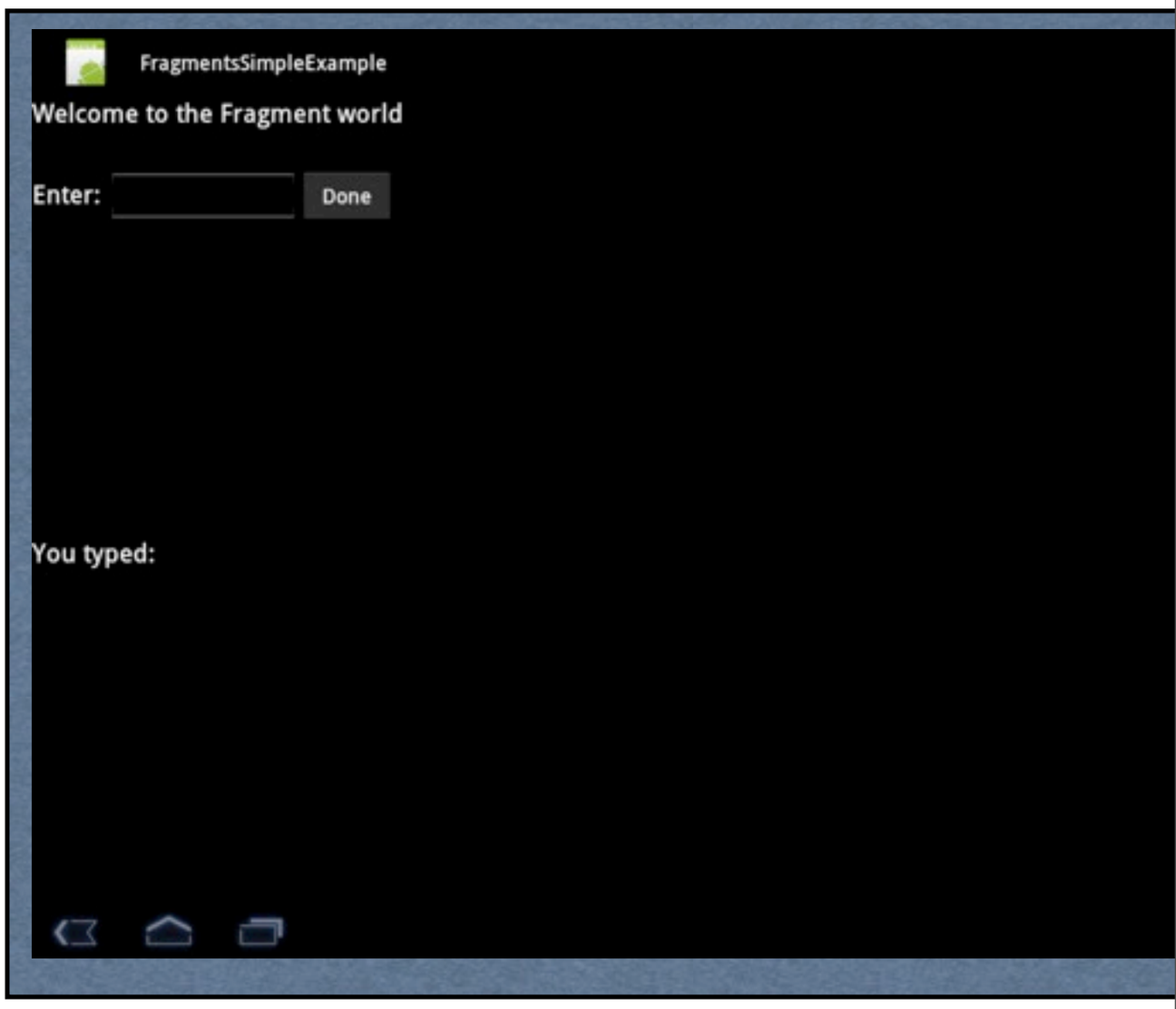
```
public static final class id {
    public static final int edit_done=0x7f050002;
    public static final int edit_fragment=0x7f050004
    public static final int edit_label=0x7f050000;
    public static final int edit_text=0x7f050001;
    public static final int result_fragment=0x7f0500
    public static final int result_label=0x7f05000
    public static final int result_text=0x7f050007;
    public static final int textView14=0x7f050003;
}
```

Fragment Example

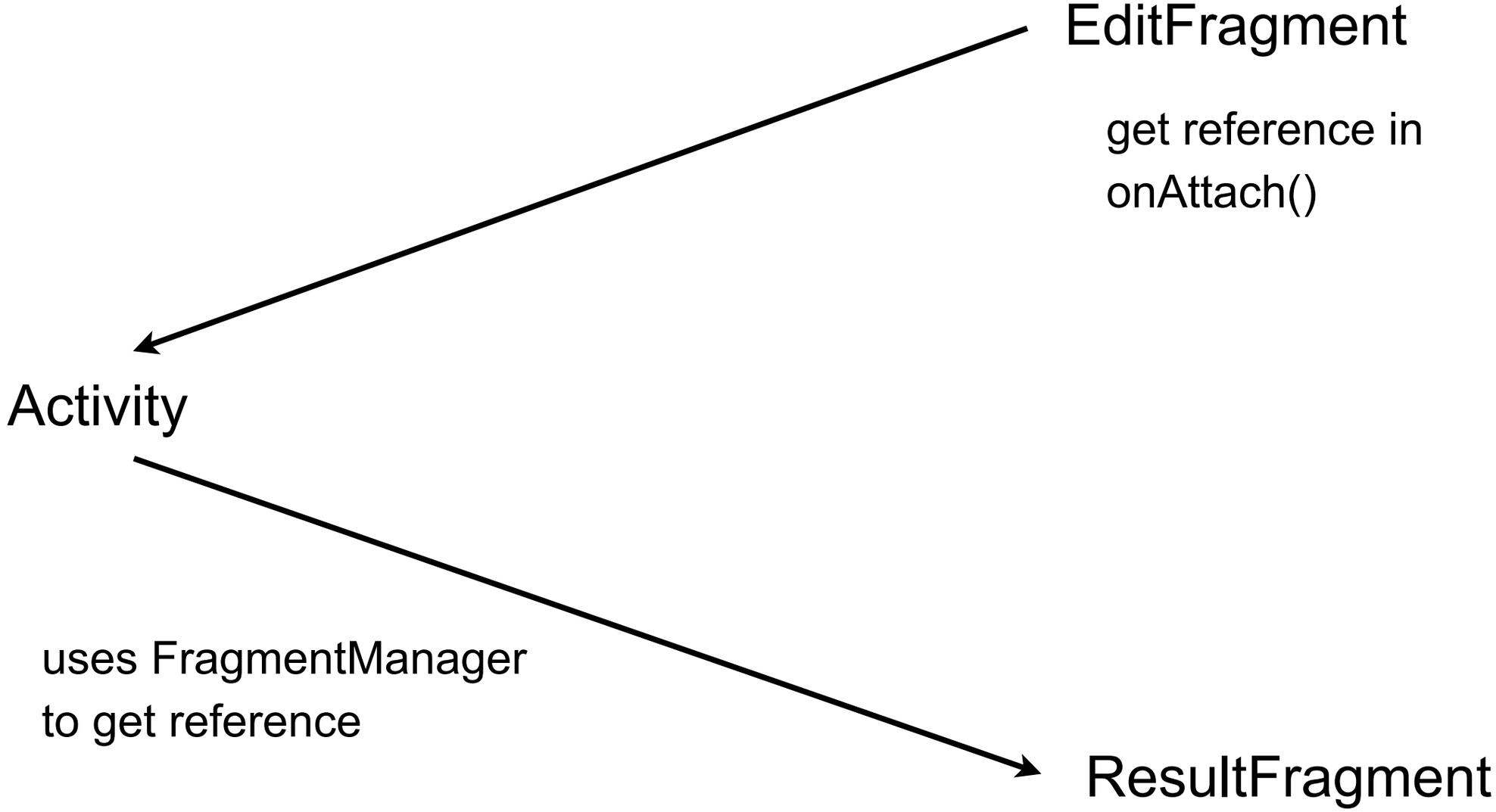
When click on "Done" button EditFragment sends text in EditField to Activity

Activity sends text to ResultFragment

ResultFragment put text in TextField



Connections



FragmentsSimpleExampleActivity

```
public class FragmentsSimpleExampleActivity extends Activity implements
OnEditFinishedListener{

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public void onEditFinished(String newText) {
      FragmentManager fragments = getFragmentManager();
      ResultFragment resultFragment
          =(ResultFragment)fragments.findFragmentById(R.id.result_fragment);
      resultFragment.setText(newText);
    }
}
```

res/layout/main.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="fill_parent"
   android:orientation="vertical" >
   <TextView
      android:id="@+id/textView14"                 android:layout_width="wrap_content"
      android:layout_height="0dp"                  android:layout_weight="1"
      android:text="Welcome to the Fragment world"
      android:textAppearance="?android:attr/textAppearanceLarge" />
 <fragment android:name="edu.sdsu.cs.whitney.EditFragment"
         android:id="@+id/edit_fragment"
         android:layout_weight="5"                android:layout_width="match_parent"
         android:layout_height="0dp" />
<fragment android:name="edu.sdsu.cs.whitney.ResultFragment"
         android:id="@+id/result_fragment"
         android:layout_weight="5"                 android:layout_width="match_parent"
         android:layout_height="0dp" />
</LinearLayout>
```

EditFragment

```
public class EditFragment extends Fragment {
    private OnEditFinishedListener editListener;
    private EditText inputText;

    public interface OnEditFinishedListener {
     public void onEditFinished(String newText);
  }

    public void done(View sourceView) {
        editListener.onEditFinished(inputText.getText().toString());
    }

    public void onAttach (Activity activity) {
        super.onAttach(activity);
        editListener = (OnEditFinishedListener) activity;
    }
```

EditFragment - Continued

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

    View editView =  inflater.inflate(R.layout.edit_fragment, container, false);

    inputText = (EditText) editView.findViewById(R.id.edit_text);

    Button done = (Button) editView.findViewById(R.id.edit_done);
     done.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
       done(v);
    }
 });
      return editView;
}
```

res/layout/edit_fragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <TextView
        android:id="@+id/edit_label"
        android:layout_width="wrap_content"     android:layout_height="wrap_content"
        android:text="Enter: "
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <EditText
        android:id="@+id/edit_text"
        android:layout_width="165dp"
        android:layout_height="wrap_content" >
        <requestFocus /></EditText>
    <Button
        android:id="@+id/edit_done"
        android:layout_width="wrap_content"      android:layout_height="wrap_content"
        android:text="Done"/>
</LinearLayout>
```

ResultFragment

```
public class ResultFragment extends Fragment {
    private TextView inputText;

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

        View resultView =  inflater.inflate(R.layout.result_fragment, container, false);
        inputText = (TextView) resultView.findViewById(R.id.result_text);
        return resultView;
    }

    public void setText(String newText) {
        inputText.setText(newText);
    }
}
```

res/layout/result_fragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
    <TextView
        android:id="@+id/result_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="You typed: "
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <TextView
        android:id="@+id/result_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

Menus & Action Bars

Fragments can add items to menus & Action Bar

in onCreate or in onCreateView call:

setHasOptionsMenu(true)

If don't do this onCreateOptionsMenu is not called

Creating menu

```
public void onCreateOptionsMenu (Menu menu, MenuInflater inflater) {
    inflater.inflate(R.menu.edit, menu);
}
```

Handling the menu

```
public boolean onOptionsItemSelected(MenuItem item) {
    process menu item here
    return (super.onOptionsItemSelected(item));
}
```

When menu item is selected by user

onOptionsItemSelected is called on fragment and activity

Does does not matter which one created menu item

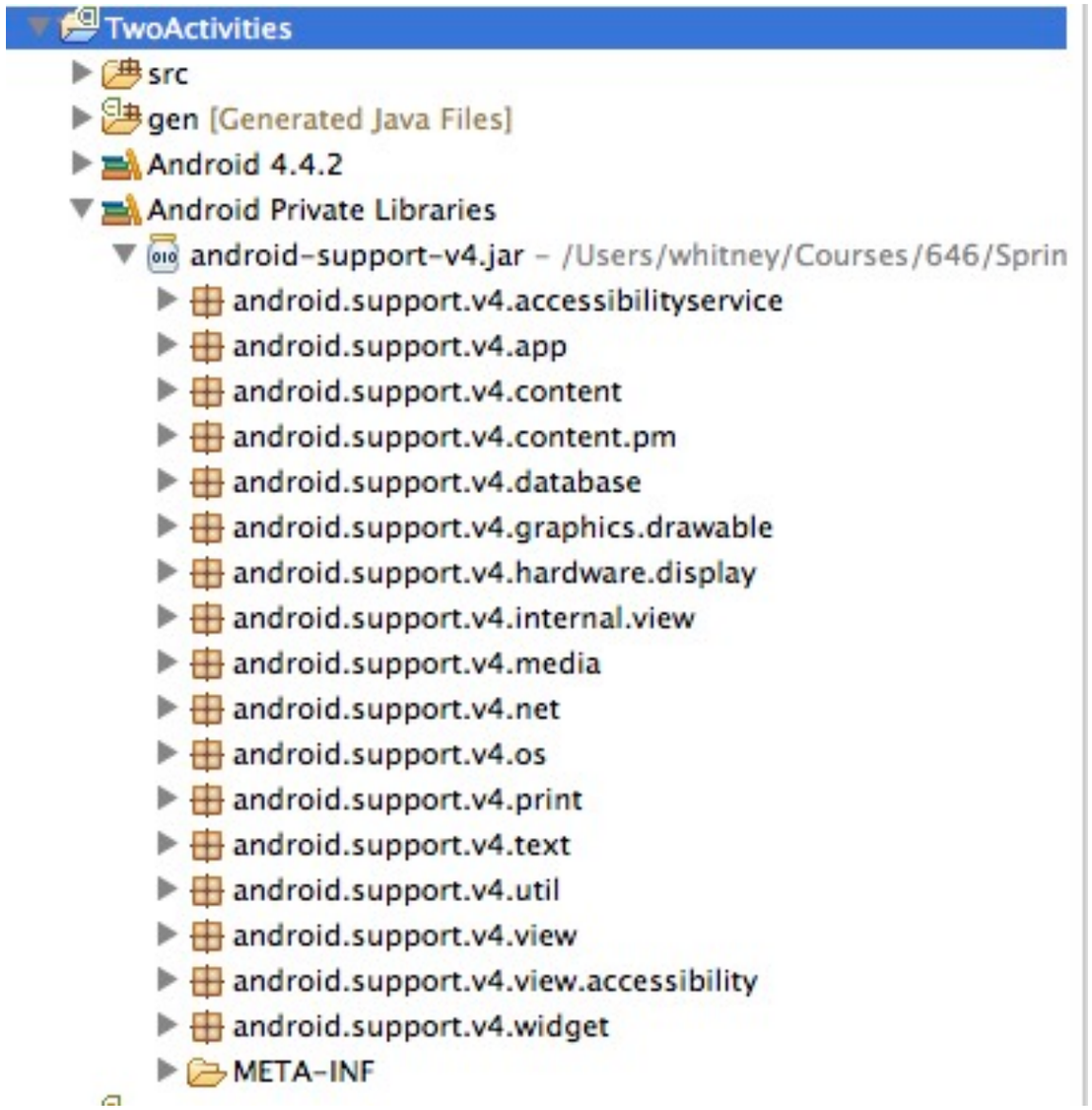
Android Compatibility Library

Back ports Fragments to Android 1.6

Some minor differences with Fragment library

Installing Android Compatibility Library

Eclipse installs Compatibility Library in your project automatically



Differences with Fragments

Subclass android.support.v7.app.Fragment not android.app.Fragment

Library does not contain
PreferenceFragment
WebViewFragment

Activity must subclass FragmentActivity

Use getSupportFragmentManager() instead of getFragmentManager()

Can not use onRetainNonConfigurationInstance() in FragmentActivity

Some Fragment Patterns

FrameLayout

Default Fragment Project uses FrameLayout

Activity layout contains just FrameLayout

Fragment layout becomes sole element in FrameLayout

static class

public static class PlaceholderFragment extends Fragment {

Default fragment project uses static Fragment class

Nested inside Activity class

Makes it harder to reuse fragment elsewhere

Empty Constructor

Each Fragment subclass needs a constructor with no arguments

```
public FragmentExample() {
    // Required empty public constructor
}
```

Used when OS needs to recreate Fragment

How does one pass data to the constructor?

setArguments

```
FragmentExample fragment = new FragmentExample();
Bundle args = new Bundle();
args.putString(“name”, “Sam”);
args.putInt(“age”, 12);
fragment.setArguments(args);
```

In FragmentExample class

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (getArguments() != null) {
        userName = getArguments().getString(“name”);
        userAge = getArguments().getString(“age”);
    }
}
```

setArguments, getArguments

Call setArguments before fragment is attached to an activity

Use getArguments in onCreate (or onAttach)

Argument bundle is retained across fragment destruction/creation

static Constructor

```
public FragmentExample {

   public static FragmentExample newInstance(String name, int age) {
         FragmentExample fragment = new FragmentExample();
         Bundle args = new Bundle();
         args.putString(“name”, name);
         args.putInt(“age”, age);
         fragment.setArguments(args);
    }

etc.
}
```