APPS

Add custom transitions to Android alert dialogs

By William J. Francis in Software Engineer, June 11, 2013, 2:03 PM PST

This Android developer demo shows how to replace the associated on and off screen transitions of an alert dialog with your own XML animation set.

Okay, I admit it: The folks at Google spoil me. The numerous helper and builder classes provided for Android development (http://developer.android.com/index.html) make routine UI tasks so simple, it's easy to overlook the deeper capabilities of the encapsulated classes.

The alert dialog and its builder are a great example. While there is nothing wrong with the default dialog, if you look beyond the basic functionality of the builder, the underlying alert dialog can be customized any number of ways prior to invoking show.

The following example demonstrates how to replace the associated on and off screen transitions of an alert dialog with your own XML animation set. You can follow along with my step-by-step instructions, or download (http://b2b.cbsimg.net/downloads/Weilage/animate_dialogs.zip) and import the entire project directly into Eclipse.

1. In Eclipse, create a new Android project. Target Android 2.2 or higher.

2. In the /res/layout folder, modify activity_main.xml to include a button and a label.

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context=".MainActivity" >
   <Button
android:id="@+id/button1"
```

```
        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentBottom="true"

        android:layout_centerHorizontal="true"

        android:layout_marginBottom="94dp"

        android:text="Show Dialog" />

    <TextView
android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_alignParentTop="true"

        android:layout_centerHorizontal="true"

        android:layout_marginTop="46dp"

        android:text="Animated Dialog Sample" />

</RelativeLayout>
```

3. In the /res folder, create a directory called "anim". We will add two files: slide_in_left.xml and slide_out_right.xml.

slide_in_left.xml

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

        android:fromXDelta="100%p" android:toXDelta="0"

        android:duration="500" />
```

slide_out_right.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<translate xmlns:android="http://schemas.android.com/apk/res/android"

    android:fromXDelta="0" android:toXDelta="100%p"

    android:duration="500" />
```

4. Open the /res/values/styles.xml file and add a new style node called dialog_animation. This is where we simply pull in the previously defined animations into a single resource we can apply to a window.

styles.xml

```
<resources>

  <style name="AppBaseTheme" parent="android:Theme.Light"/>
<style name="AppTheme" parent="AppBaseTheme"/>

   <style name="dialog_animation">
<item name="android:windowEnterAnimation">@anim/slide_in_left</item>

<item name="android:windowExitAnimation">@anim/slide_out_right</item>

</style>

</resources>
```

5. Now we are ready to write some code. Open the /src/MainActivity.java file. Most of the code should look familiar. We extend the Activity class, implement the OnClickListener, and override both the onCreate and onDestroy. Make note of the single line of code in the onClick handler immediately preceding the call to dialog.show() -- this is where we apply our customization prior to showing the dialog.

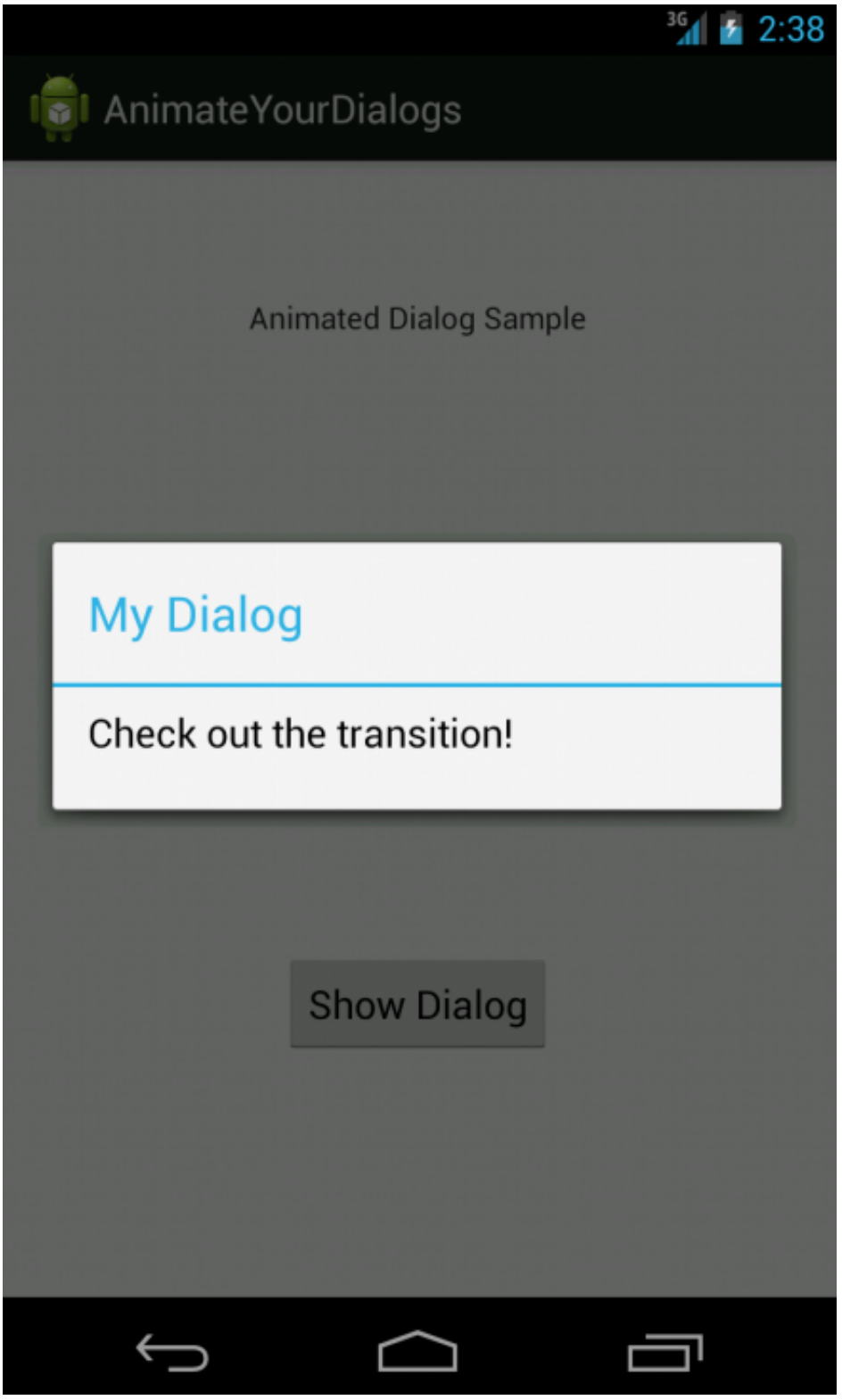
MainActivity.java

```
package com.authorwjf.animateyourdialogs;

import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.app.Activity;
import android.app.AlertDialog;

public class MainActivity extends Activity implements OnClickListener {

    AlertDialog dialog;

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);


findViewById(R.id.button1).setOnClickListener(this);
}

    @Override
    public void onClick(View v) {
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("My Dialog");


builder.setMessage("Check out the transition!");


dialog = builder.create();


dialog.getWindow().getAttributes().windowAnimations =


R.style.dialog_animation;


dialog.show();

    }

    @Override
    public void onDestroy() {
        if (dialog!=null) {
            if (dialog.isShowing()) {
dialog.dismiss();


}


}


        super.onDestroy();
}


}
```

When you run the code, you'll immediately notice how the alert dialog now slides in from the left, then back out to the right (**Figure A**). Is this an improvement over the default animation for an alert dialog? Probably not. But the point is it's not necessary to think of the builders found in the framework and the underlying classes they build as

mutually exclusive. Use the builders as a starting point, but take the time to explore the classes being built as well. Understanding both will allow you to take advantage of the provided helpers, without sacrificing the ability to tweak your user experience. **Figure A**



(http://b2b.cbsimg.net/blogs/custom_alert_dialog_anim.png)

About William J. Francis

William J Francis began programming computers at age eleven. Specializing in embedded and mobile platforms, he has more than 20 years of professional software engineering under his belt, including a four year stint in the US Army's Military Intellige...

Recommended

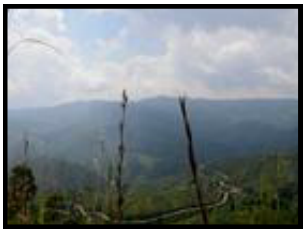
Add your Comment

Editor's Picks

We-commerce: The sharing economy's uncertain path

'PayPal Mafia' redefined success in Silicon Valley

How Aaron Levie and his childhood friends built Box



Conflict minerals funded a war that killed millions