



LittleBox
Solutions

A Crash Course: Widgets and Services

James Puderer
October 19th, 2010

Overview

- Create a virtual cat that will live on the desktop
- The application will consist of two main components:
 - Widget – Responsible for drawing the cat
 - Service – Responsible for the cat's behaviour
- Widget does not require an activity



Step #1 – Layout the Widget

```
<?xml version="1.0" encoding="utf-8"?>
<ImageView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/widget_cat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:src="@drawable/cat"/>
```

- Not all views and layouts can be used
- Limited to those supported by *RemoteViews*
- Remote views describe view hierarchies that can be displayed in other process.
- Widgets are displayed in the *Launcher* activity.

Step #2 – Describe the Widget

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="72dip"
    android:minHeight="72dip"
    android:initialLayout="@layout/widget_layout"
    android:updatePeriodMillis="0"
/>
```

- Create XML file in res/xml directory
- Specify widget size
 - Rounded up to the nearest standard widget size
 - Standard size depends on orientation
- Points to widget layout
- Update period determine how often to call *onUpdate* method.

Step #3 – Create the Widget

```
public class CatWidget extends AppWidgetProvider {  
    public void onUpdate(Context context,  
                        AppWidgetManager appWidgetManager,  
                        int[] appWidgetIds) {  
        // Do nothing  
    }  
}
```

- Absolute minimum widget.
- Does nothing except display (cat picture)

Step #4 – Update the Manifest

```
<!-- Broadcast Receiver that will process AppWidget updates -->
<receiver android:name=".CatWidget" android:label="@string/widget_name">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/widget" />
</receiver>
```

- Declare our AppWidget as a broadcast receiver
 - AppWidgets are a type of broadcast receiver
- Add the meta-data tag to point to the widget XML file
 - This is how Android "sees" the widget

Test Break

- You should now be able to compile the application
- Widget can be added to home screen
- Widget doesn't yet do anything
- *It could be considered cruel to leave your cat in the desert!*



Step #5 – Create the Service

```
public class CatService extends Service {  
    ...  
    @Override public void onCreate() { ... }  
    @Override public void onDestroy() { ... }  
    @Override public IBinder onBind(Intent intent) { ... }  
    @Override public int onStartCommand(Intent intent,  
        int flags, int startId) { ... }  
    ...  
}
```

- Service controls cat's behaviour
 - Cat meows when it is needy (unhappy)
 - Catt purrs when it is happy
- Receives intent whenever widget is touched. Touching makes the cat happier.
- Receives alarm intent at regular interval. The alarm intent makes the cat less happy.

Step #6 – Update the Widget

```
public class CatWidget extends AppWidgetProvider {  
    public void onUpdate(Context context,  
                          AppWidgetManager appWidgetManager,  
                          int[] appWidgetIds) {  
        ...  
        // ADD STUFF HERE!  REFER TO SOURCE CODE  
        ...  
    }  
}
```

- Start the service
- Create a PendingIntent to that gets sent whenever the widget is clicked on
- *Note: The onUpdate method is called on widget creation, but **would not** be called if we had defined a configuration activity for the widget.*

Step #7 – Update the Manifest

```
<!-- Service to control cat's behaviour -->  
<service android:name=".CatService" />
```

- Add entry for the *CatService*

That's it!

■ Caveats

- Pending intents are reused if intent has the same target. The comparison code does not take 'extra' data into account (See the comment labelled *HACK* in the source)
- Putting a ringer on vibrate or silent does not affect the media volume
- Kill the service when there are no more widgets
- Kill the alarm when the service gets destroyed

■ Extra Credit

- Add a configuration activity for the widget to control the cat's behaviour
- Add a broadcast receiver and get the cat to respond to external events

References

- **Android Blog: Introducing home screen widgets and the AppWidget framework**
<http://android-developers.blogspot.com/2009/04/introducing-home-screen-widgets-and.html>
- **Android Developer Guide: App Widgets**
<http://developer.android.com/guide/topics/appwidgets/index.html>
- **Android Developer Guide: Widget Design Guidelines**
 - *http://developer.android.com/guide/practices/ui_guidelines/widget_design.html*