

THE BROADCASTRECEIVER CLASS

BROADCASTRECEIVER

Base class for components that receive and react to events

BROADCASTRECEIVER

BroadcastReceivers register to receive events in which they are interested

BROADCAST RECEIVER

When Events occur they are represented as Intents

Those Intents are then broadcast to the system

BROADCAST RECEIVER

Android routes the Intents to BroadcastReceivers that have registered to receive them

BroadcastReceivers receive the Intent via a call to `onReceive()`

TYPICAL USE CASE

Register BroadcastReceivers

Broadcast an Intent

Android delivers Intent to
registered recipients by calling
their `onReceive()` method

Event handled in `onReceive()`

REGISTERING FOR INTENTS

BroadcastReceivers can register in two ways

Statically, in AndroidManifest.XML

Dynamically, by calling a
registerReceiver() method

STATIC REGISTRATION

Put <receiver> and <intent-filter> tags in AndroidManifest.xml

<RECEIVER> FORMAT

<receiver

android:enabled=["true" | "false"]

android:exported=["true" | "false"]

android:icon=~~draw~~drawable resource

android:label=~~string~~string resource

android:name=~~string~~string

android:permission=~~string~~string

android:process=~~string~~string >

. . .

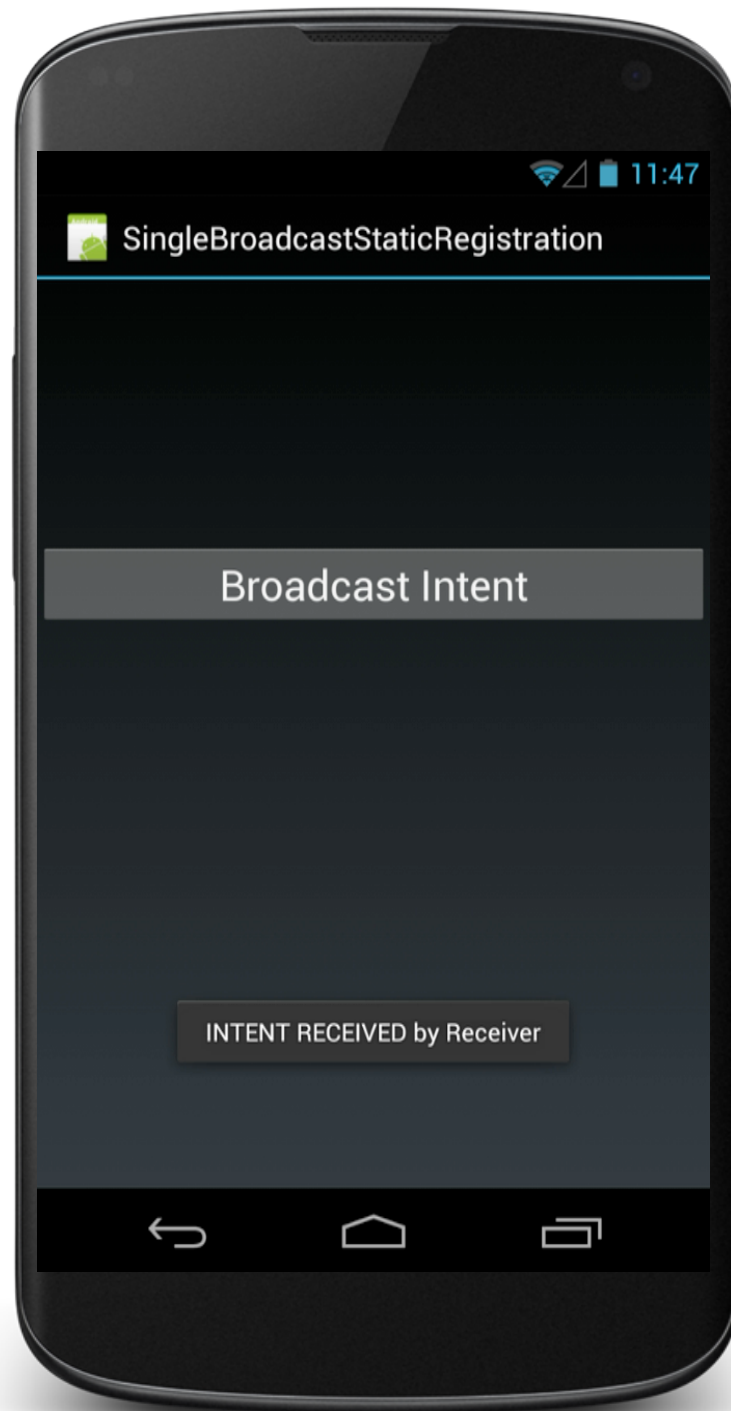
</receiver>

INTENT FILTER

Specify <intent-filter> tag within the <receiver>

STATIC REGISTRATION

Receivers are registered with the system at boot time or when their application package is added at runtime



BroadcastReceiver Static Registration

```
Receiver.java SimpleBroadcast.java ✕
1 package course.examples.BroadcastReceiver.singleBroadcastStaticRegistration;
2
3+ import android.app.Activity;
9
10 public class SimpleBroadcast extends Activity {
11
12     private static final String CUSTOM_INTENT = "course.examples.BroadcastReceiver.show_toast";
13
14 @Override
15 public void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.main);
18
19     Button button = (Button) findViewById(R.id.button);
20 button.setOnClickListener(new OnClickListener() {
21     @Override
22     public void onClick(View v) {
23
24         sendBroadcast(new Intent(CUSTOM_INTENT),
25             android.Manifest.permission.VIBRATE);
26     }
27 });
28 }
29 }
```

BroadcastReceiverSingleBroadcastStaticRegistration

Receiver.java

SimpleBroadcast.java

```
1 package course.examples.BroadcastReceiver.singleBroadcastStaticRegistration;
2
3 import android.content.BroadcastReceiver;
4
5
6 public class Receiver extends BroadcastReceiver {
7
8     private final String TAG = "Receiver";
9
10    @Override
11    public void onReceive(Context context, Intent intent) {
12
13        Log.i(TAG, "INTENT RECEIVED");
14
15        Vibrator v = (Vibrator) context
16            .getSystemService(Context.VIBRATOR_SERVICE);
17        v.vibrate(500);
18
19        Toast.makeText(context, "INTENT RECEIVED by Receiver", Toast.LENGTH_LONG).show();
20
21    }
22 }
23
24
25
26
27 }
```

DYNAMIC REGISTRATION

Create an IntentFilter

Create a BroadcastReceiver

Register BroadcastReceiver using
registerReceiver()

LocalBroadcastManager

Context

Call unregisterReceiver() to
unregister BroadcastReceiver

BroadcastReceiver.BroadcastReceiver

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    mBroadcastMgr = LocalBroadcastManager
        .getInstance(getApplicationContext());
    mBroadcastMgr.registerReceiver(receiver, intentFilter);

    setContentView(R.layout.main);

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            mBroadcastMgr.sendBroadcast(new Intent(CUSTOM_INTENT));
        }
    });
}
```


EVENT BROADCAST

Several broadcast methods supported

Normal vs. Ordered

Normal: processing order undefined

Ordered: sequential processing in priority order

EVENT BROADCAST

Sticky vs. Non-Sticky

Sticky: Store Intent after initial broadcast

Non-Sticky: Discard Intent after initial broadcast

With or without receiver permissions

EVENT DELIVERY

Intents delivered by calling
`onReceive()`, passing in:

the Context in which the receiver is
running

the Intent that was broadcast

EVENT HANDLING IN onReceive()

Hosting process has high priority
while onReceive() is executing

EVENT HANDLING IN ONRECEIVE()

onReceive() runs on the main Thread, so it should be short-lived

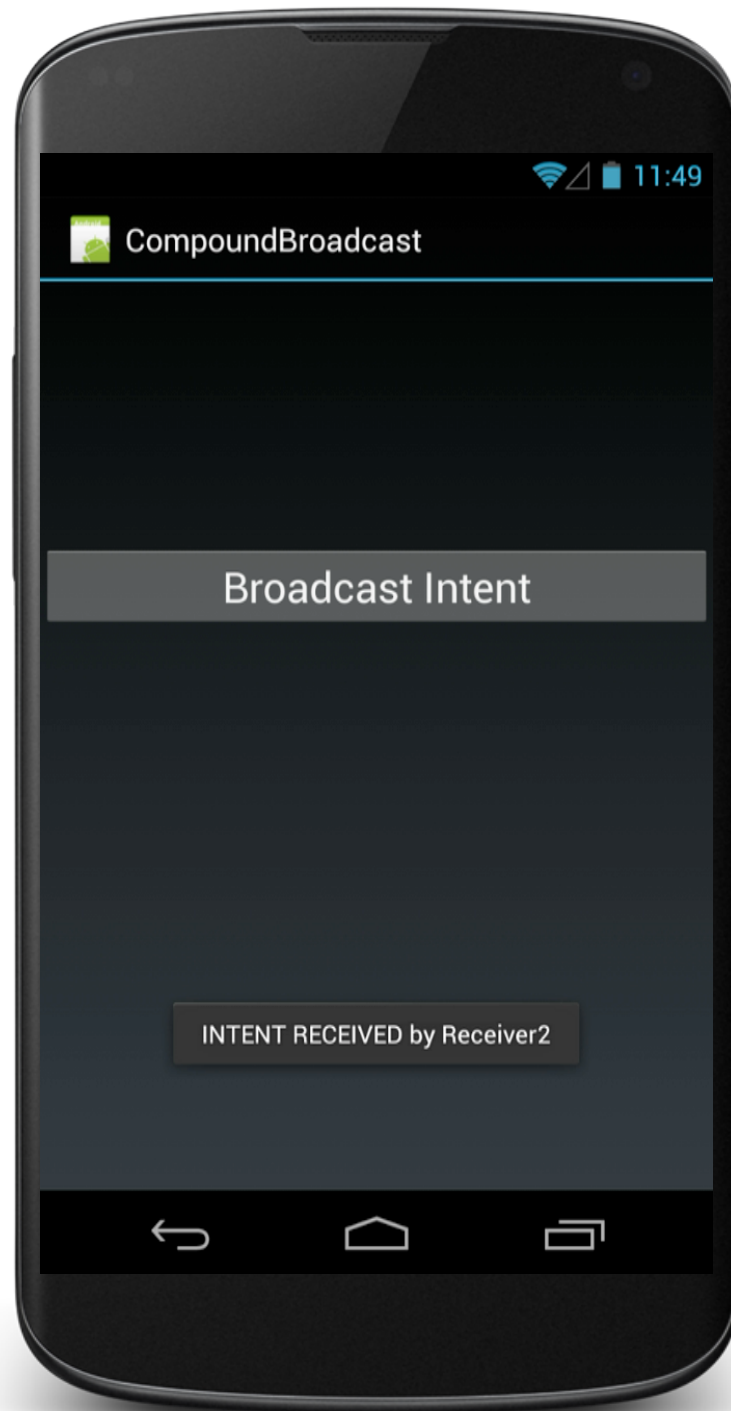
If event handling is lengthy, consider starting a Service, rather than performing complete operation in onReceive()

EVENT HANDLING IN ONRECEIVE()

Receiver is not considered valid once `onReceive()` returns

Normally `BroadcastReceivers` can't start asynchronous operations

e.g., showing a dialog, starting an Activity via `startActivityForResult()`



ORDERED BROADCASTS

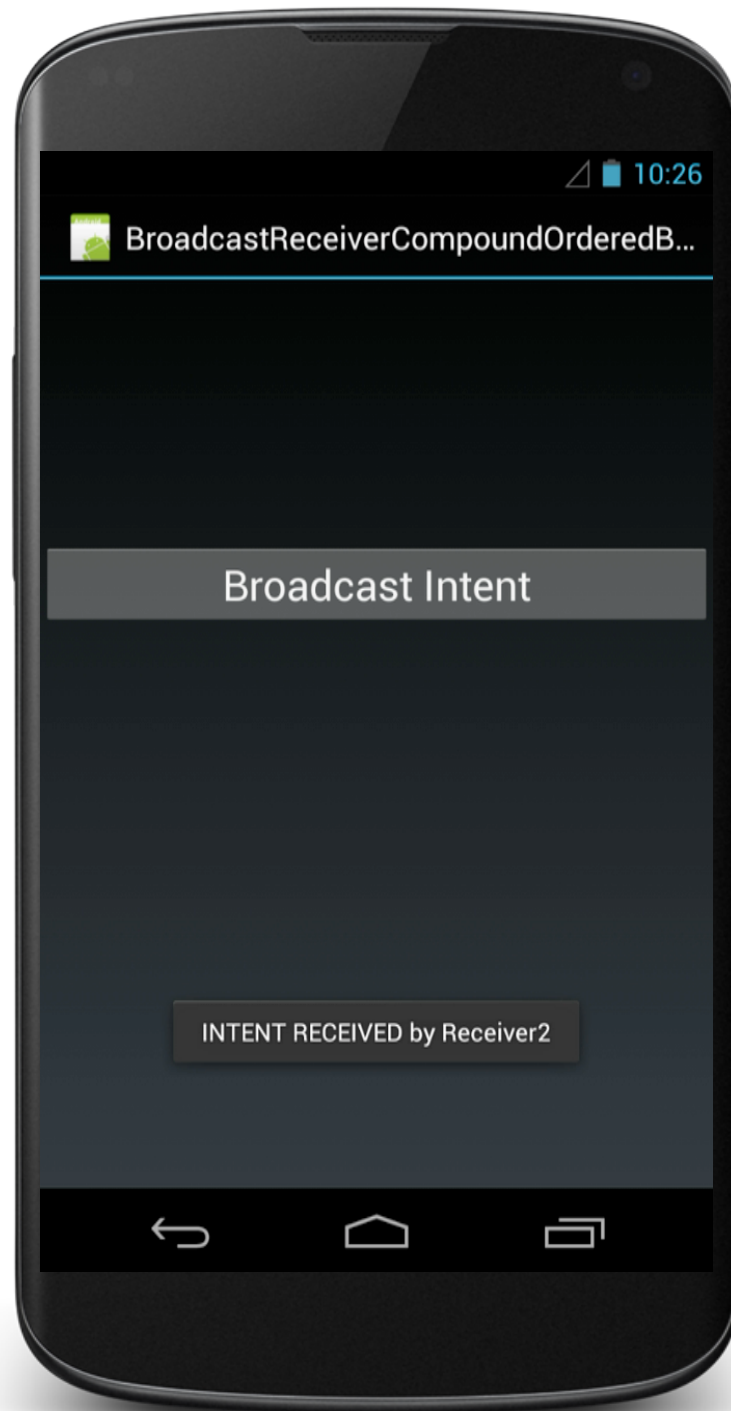
// send Intent to BroadcastReceivers in priority order

```
void sendOrderedBroadcast (Intent intent, #  
                           String receiverPermission)
```

// send Intent to BroadcastReceivers in priority order

// includes multiple parameters for greater control

```
void sendOrderedBroadcast (Intent intent, #  
                           String receiverPermission, #  
                           BroadcastReceiver resultReceiver,  
                           Handler scheduler, #  
                           int initialCode, #  
                           String initialData, #  
                           Bundle initialExtras)
```

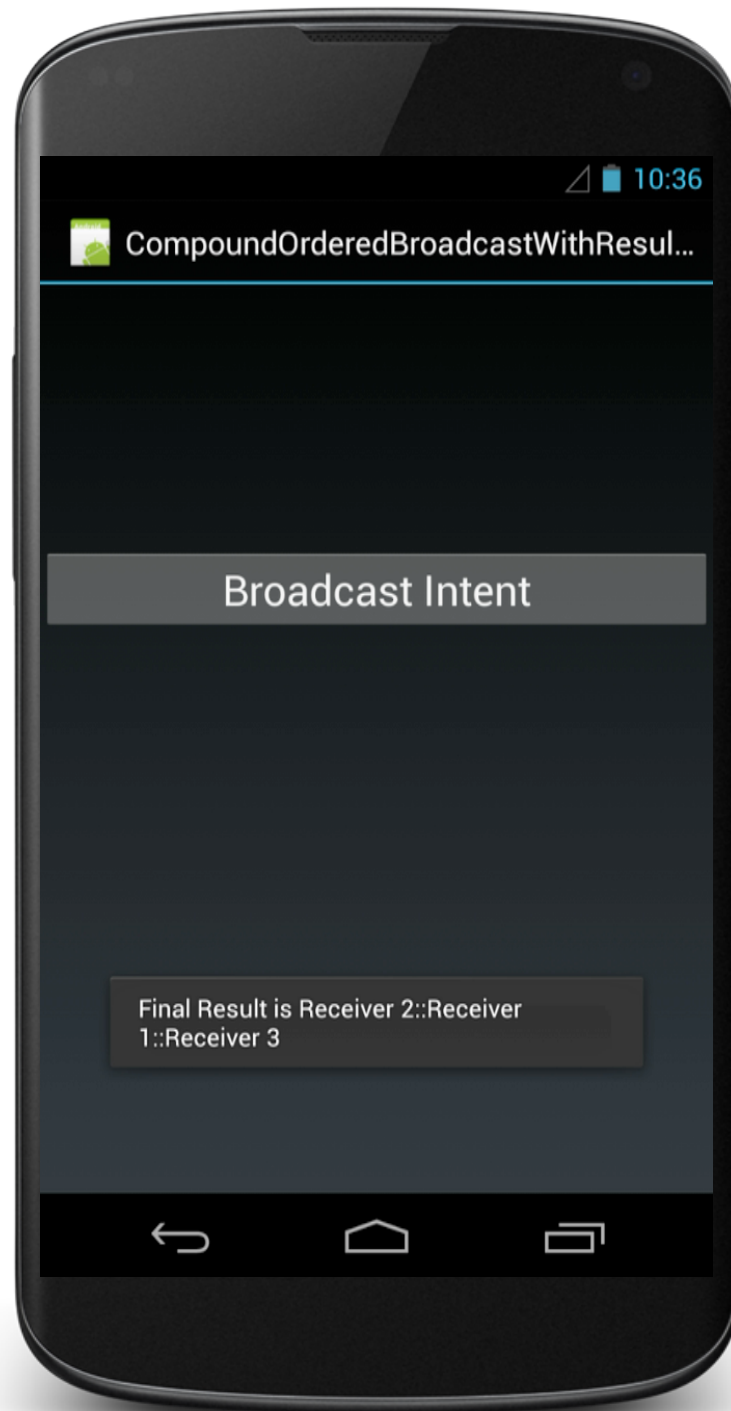



BroadcastReceiver

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

    IntentFilter intentFilter = new IntentFilter(CUSTOM_INTENT);
    intentFilter.setPriority(3);
    registerReceiver(mReceiver, intentFilter);

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            sendOrderedBroadcast(new Intent(CUSTOM_INTENT),
                android.Manifest.permission.VIBRATE);
        }
    });
}
```



BroadcastReceiver

```
private final Receiver1 mReceiver1 = new Receiver1();

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.main);

    IntentFilter intentFilter = new IntentFilter(CUSTOM_INTENT);
    intentFilter.setPriority(3);
    registerReceiver(mReceiver1, intentFilter);

    Button button = (Button) findViewById(R.id.button);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            sendOrderedBroadcast(new Intent(CUSTOM_INTENT), null,
                new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent intent) {
                        Toast.makeText(context,
                            "Final Result is " + getResultData(),
                            Toast.LENGTH_LONG).show();
                    }
                }, null, 0, null, null);
        }
    });
}
```

STICKY BROADCASTS

Sticky Intents are cached by
Android

New Intents overwrite older Intents
they match

STICKY BROADCASTS

When BroadcastReceivers are dynamically registered

Cached sticky Intents matching the specified IntentFilter are broadcast to the BroadcastReceiver

One matching sticky Intent is returned to the caller

STICKY BROADCASTS

```
//public abstract class Context ...
```

```
// send sticky Intent to interested BroadcastReceivers
```

```
void sendStickyBroadcast (Intent intent)
```

```
// send sticky Intent to interested BroadcastReceivers in priority order
```

```
// sender can provide various parameters for greater control
```

```
void sendStickyOrderedBroadcast (Intent intent,          #  
                                BroadcastReceiver resultReceiver, #  
                                Handler scheduler, #  
                                int initialCode, #  
                                String initialData, #  
                                Bundle initialExtras)
```

Broadcaster must have BROADCAST_STICKY
permission to send sticky Intents

BroadcastReceiver

```
registerReceiver(new BroadcastReceiver() {  
    @Override  
    public void onReceive(Context context, Intent intent) {  
  
        if (intent.getAction().equals(Intent.ACTION_BATTERY_CHANGED)) {  
  
            String age = "Reading taken recently";  
            if (isInitialStickyBroadcast()) {  
                age = "Reading may be stale";  
            }  
  
            currentStateView.setText("Current Battery Level:"  
                + String.valueOf(intent.getIntExtra(  
                    BatteryManager.EXTRA_LEVEL, -1))  
                + System.getProperty("line.separator") + age);  
        }  
    }  
}, new IntentFilter(Intent.ACTION_BATTERY_CHANGED));
```