

Intent – Broadcast and Receive Messages

Note: To set up a broadcast receiver, include the following in the activity from which you want to listen to broadcast:

```
public class MyBroadcastReceiver extends BroadcastReceiver {  
    @Override  
    public void onReceive(Context context, Intent i) {  
        // access the broadcast message through i.getStringExtra("key")  
    }  
}
```

Note: Include also the following in the activity from which you want to listen to broadcast:

```
MyBroadcastReceiver myReceiver;  
IntentFilter         myIntentFilter;
```

In the onCreate method, add:

```
myReceiver      = new MyBroadcastReceiver();  
myIntentFilter  = new IntentFilter("SOC_Broadcast");
```

Register the receiver in onResume() :

```
registerReceiver(myReceiver, myIntentFilter);
```

Unregister the receiver in onPause():

```
unregisterReceiver(myReceiver);
```

To send a broadcast (can be from a different activity):

```
Intent i = new Intent("SOC_Broadcast");  
i.putExtra("key", "some value from intent");  
sendBroadcast(i);
```

```
@Override  
public void onResume() {  
    super.onResume();  
    //-- register the receiver --  
    registerReceiver(myReceiver, myIntentFilter);  
}
```

```
@Override  
public void onPause() {  
    super.onPause();  
    //-- unregister the receiver --  
    unregisterReceiver(myReceiver);  
}
```

Can also define a receiver through the AndroidManifest.xml file:

```
<receiver android:name=".SecondReceiver">
    <intent-filter>
        <action android:name="SOC_Broadcast"/>
    </intent-filter>
</receiver>
```

Then define a SecondReceiver class to listen to broadcast:

```
public class SecondReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent i) {
        // access the received message through i.getStringExtra("key")
    }
}
```

Note: Broadcast receivers registered through AndroidManifest.xml are useful for broadcasts that may go on while your code is not in memory (e.g., BOOT_COMPLETED, scheduled alarms via AlarmManager etc.).

To read the status of battery:

```
IntentFilter ifilter =  
    new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
```

```
Intent batteryStatus = this.registerReceiver(null, ifilter);
```

because it is a sticky intent, no need to register a Broadcast Receiver. Can just pass in a null here

```
// Is battery charging or full?
```

```
int status = batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);  
boolean isCharging = status == BatteryManager.BATTERY_STATUS_CHARGING ||  
    status == BatteryManager.BATTERY_STATUS_FULL;
```

```
// How are we charging the battery?
```

```
int chargePlug = batteryStatus.getIntExtra(BatteryManager.EXTRA_PLUGGED, -1);  
boolean usbCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_USB;  
boolean acCharge = chargePlug == BatteryManager.BATTERY_PLUGGED_AC;
```

```
// check battery level
```

```
int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);  
int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);  
float batteryPct = level / (float)scale;
```