Programming
Handheld Systems

Adam Porter
The BroadcastReceiver Class
Today's Topics

The Broadcast Class
Registration
Broadcast Processing
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 BroadcastReceiver that have registered to receive them

Broadcast Receivers receive the Intent via a call to onReceive()
Typical Use Case

Register BroadcastReceiver

Broadcast an Intent

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

Event handled in onReceive()
Registering for Intents

Broadcast Receivers 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="drawable resource"
    android:label="string resource"
    android:name="string"
    android:permission="string"
    android:process="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.
Dynamic Registration

Create an IntentFilter
Create a BroadcastReceiver
Register BroadcastReceiver using registerReceiver()

LocalBroadcastManager Context

As necessary, call unRegisterReceiver() to unregister BroadcastReceiver
Event Broadcast

Several broadcast methods supported

Normal vs. Ordered

Normal: processing order undefined
Ordered: sequential processing in priority order

Sticky vs. Non-Sticky

Sticky: Store Intent after initial broadcast
Non-Sticky: Discard Intent after initial broadcast

With or without receiver permissions
Some debugging tips

Log extra Intent resolution information

Intent.setFlag(FLAG_DEBUG_LOG_RESOLUTION)

List registered BroadcastReceiver

Dynamically registered

  % adb shell dumpsys activity b

Sectatically registered

  % adb shell dumpsys package
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 Broadcast Receivers can’t start asynchronous operations**

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

**Can use `goAsync()` to keep Broadcast Receiver alive**
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)
Sticky Broadcasts

Sticky Intents are cached by Android

New Intents overwrite older Intents they match

When Broadcast Receivers are dynamically registered

Cached sticky Intents matching the specified IntentFilter are broadcast to the Broadcast Receiver

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
Next Time

Threads, AsyncTasks & Handlers