Recording in Progress

This class is being recorded

Please turn off your video and/or video if you do not wish to be recorded
CMSC436: Programming Handheld Systems
The BroadcastReceiver Class
Today’s Topics

The BroadcastReceiver Class
Registering for events
Broadcasting events
Processing events
BroadcastReceiver

Base class for components that receive and react to events
BroadcastReceiver

BroadcastReceivers register to receive events in which they are interested
BroadcastReceiver

When Events occur at runtime they are represented as Intents
Those Intents are then broadcast to the system
BroadcastReceiver

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 Broadcast Receivers to receive specific events

When event occurs, 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>
  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 a <receiver>
See lecture on Intent class
Static Registration

Receivers can be registered in AndroidManifest.xml
Will be woken to receive broadcasts, if needed
In API 26+, statically registered receivers cannot receive most implicit intents

BcastRec
SinBcast
StatReg

MainActivity: Broadcast sent.
Receiver: Broadcast received.

A Toast from Receiver!
Dynamic Registration

Create an IntentFilter
Create a BroadcastReceiver
Register using Context.registerReceiver()
Unregister using Context.unregisterReceiver()
BcastRec
CompBcast

Receiver1: Intent received.
Receiver3: Intent received.
Receiver2: Intent received.
Event Broadcast

Multiple broadcast methods supported

Normal vs. Ordered

Normal: processing order undefined

Ordered: sequential processing in priority order
Some Debugging Tips

Log extra Intent resolution information
   Intent.setFlag(FLAG_DEBUG_LOG_RESOLUTION)

List registered BroadcastReceivers

Dynamically registered
   % adb shell dumpsys activity b

Statically registered
   % adb shell dumpsys package
Event Delivery

Intents are delivered to BroadcastReceiver by calling `onReceive(Context, Intent)`

- 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
onReceive() runs on the main Thread
So onReceive() should be short-lived
Event Handling in onReceive()

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

Will cover the Service class later in the course
Event Handling in onReceive()

BroadcastReceiver is not considered valid once onReceive() returns

Normally, BroadcastReceivers can’t start asynchronous operations

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

Why not?
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)
BcastRecCompOrd
BcastWithResRec
Long-Running Operations

After `onReceive()` exits, system can kill `BroadcastReceiver`

Don’t’ start long-running Threads from `onReceive()`

Options

Call `goAsync()`

Schedule a JobService with JobScheduler. (Will discuss Services later in course)
goAsync()

BroadcastReceiver is generally valid only until onReceive() exits

Use goAsync() to allow asynchronous processing from onReceive()

Method returns an object of PendingResult

Receiver considered alive until PendingResult.finish()
Additional Notes

BroadcastReceiver’s original design has changed to improve security, performance and UX

- Prefer LiveData, etc. to broadcasts within an app
- Prefer Context registration over Manifest registration
- Don’t put sensitive info in implicit Intents you broadcast
- Don’t start Activities from onReceive()
Next Time

Firebase
Example Applications

BcastRecSinBcastStatReg
BcastRecSinBcastDynReg
BcastRecCompBcast
BcastRecCompOrdBcast
BcastRecCompOrdBcastWithResRec
BcastRecGoAsync