# Broadcast总结

目录

[Broadcast总结 1](#_Toc43582374)

[一、Broadcast机制简介 2](#_Toc43582375)

[二、广播分类 2](#_Toc43582376)

[三、接收广播消息 3](#_Toc43582377)

[四、发送自定义广播 4](#_Toc43582378)

[五、本地广播 4](#_Toc43582379)

[六、其他 4](#_Toc43582380)

## 一、Broadcast机制简介

广播是Android的消息机制之一，是为了便于进行系统级别的消息通知而引

入的。程序可以对自己感兴趣的广播进行注册，这样该程序就会接收到自己所关心的广播内容。这些广播可能来自系统，也可以是其他程序。Android中有完整的API用于发送和接收广播。发送广播使用Intent，接收广播使用Broadcast Receiver(广播接收器)。

## 二、广播分类

Android中的广播主要分为两种：标准广播和有序广播。

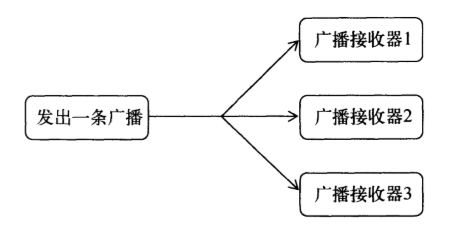
* 标准广播（Normal broadcasts）

完全异步执行的广播，在广播发出后，所有广播接收器几乎都会在同一时刻

接收到这条广播消息，因此它们之间没有任何先后顺序。

这种广播效率比较高，同时无法被截断。

工作流程如下：

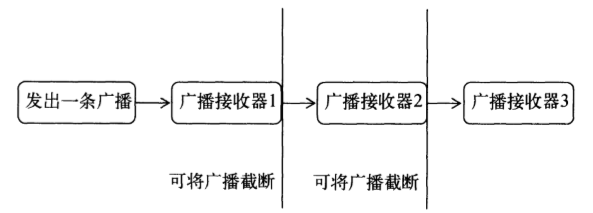


* 有序广播（Ordered broadcasts）

同步执行的广播，在广播发出后，同一时刻只会有一个广播接收器能够收到

这条广播消息，当这个广播接收器中的逻辑执行完毕后，广播才会继续传递。这种类型的广播接收器有先后顺序，优先级高的先接受广播消息，并可以截断广播。

工作流程如下：



## 三、接收广播消息

Android内置了很多系统级别的广播，比如手机开机完成、电量变化等情况，

都会发出广播。可以在程序中接收这些广播。

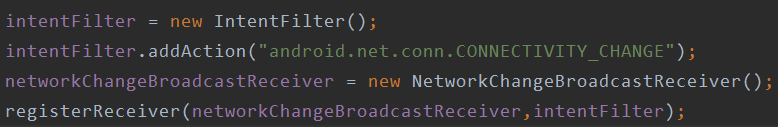
广播接收器通过注册来监听广播，有两种注册方式：动态注册和静态注册

3.1动态注册

核心方法：registerReceiver(BroadcastReceiver br,IntentFilter i),

IntentFilter指明要接收的广播

BroadcastReceiver指定广播接收器，接收到广播后，在广播接收器的onReceive中进行逻辑处理。



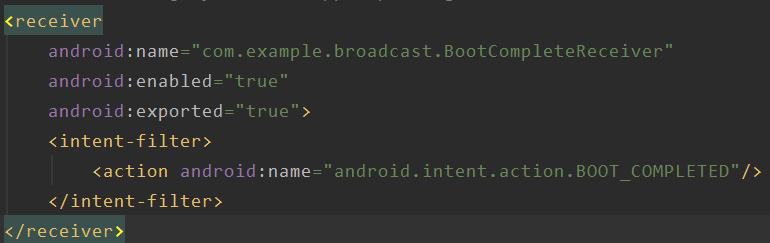
广播使用后，需要解除注册。



动态注册的广播接收器可以自由地控制注册与注销，更加灵活，但它必须在程序启动后才能接收到广播。如果要在程序未启动的情况下接收广播，则需要静态注册。

3.2静态注册

也是通过action来确定要接收的广播。



## 四、发送自定义广播

4.1发送标准广播

核心方法：sendBroadcast(Intent intent)。

Intent定义当前广播的action：

Intent intent = new Intent(“com.example.mybroadcast.Action”)

4.2发送有序广播

核心方法：sendOrderedBroadcast(Intent i，String s)。

第一个参数Intent，定义action

第二个参数String，权限相关字符串，默认设为null。

有序广播的顺序：

通过广播接收器的<intent-filter android:priority=”100”>属性设置，值越大越靠前。

广播截断：

abortBroadcast();

## 五、本地广播

前面的广播都属于全局广播，广播中使用intent进行数据传递，为了保证数据的安全性，并避免接收到其他程序发送的无用广播，Android引入了本地广播机制。通过此机制发出广播只会在程序内部传递，并且广播接收器只会接收此程序发出的广播，安全问题就解决了。

本地广播核心：LocalBroadcastManager。

它同样提供了发送广播和注册广播接收器的方法。使用方法和全局广播一样。

本地广播不能通过静态注册的方式来接收。因为静态注册主要就是为了让程序在未启动的情况下也能收到广播，而发送本地广播时，程序已经启动。

本地广播优势：

* 广播只在程序内部传播，数据不会泄露。
* 其他程序无法发送广播到程序内部，没有安全漏斗的隐患。
* 发送本地广播比发送全局广播更高效。

## 六、其他

在广播接收器的onReceive()中，不能添加过多逻辑或者进行耗时操作，因为在广播接收器中不允许开启线程，当onReceive()运行较长时间而没有结束，程序就好报错。广播接收器更多的扮演打开程序其他组件的角色，比如创建状态栏通知，或者启动一个服务等。