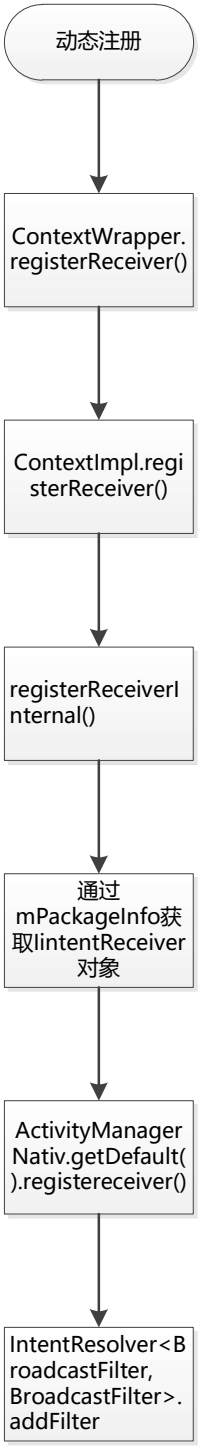
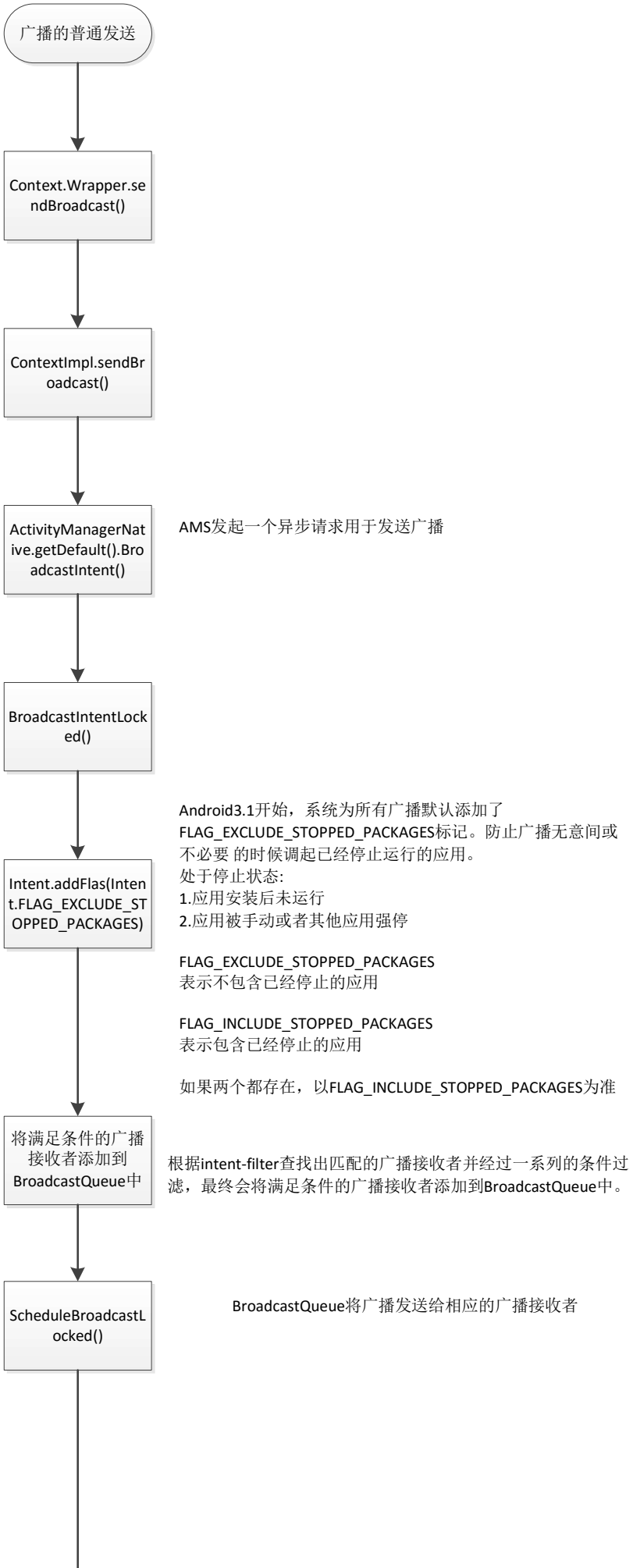
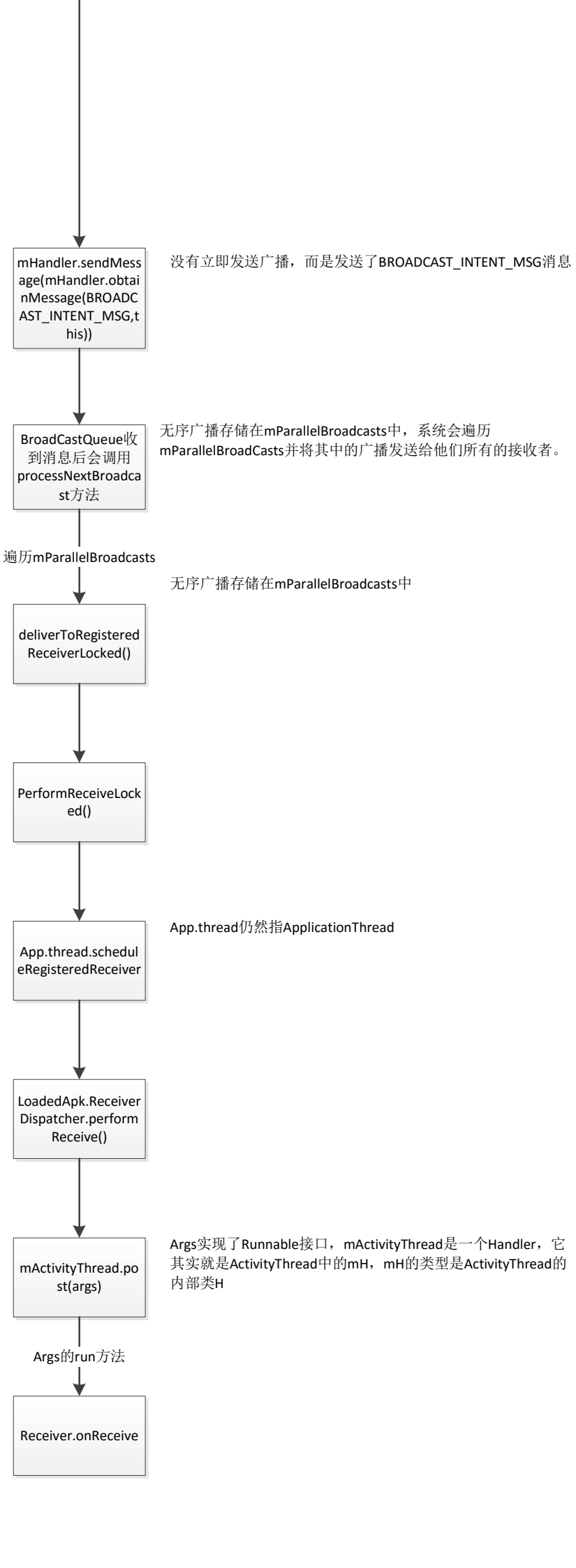

动态注册
ContextWrapper.
registerReceiver()
ContextImpl.regi
sterReceiver()
registerReceiverI
nternal()
通过
mPackageInfo获
取IintentReceiver
对象
1.进程间通信，所以要将BroadcastReceiver转化为
IintentReceiver (是一个Binder)
2.IintentReceiver是
LoadedApk.ReceiverDispatcher.InnerReceiver
3.ReceiverDispatcher内部同时保存了BroadcastReceiver和
InnerReceiver，这样当接收到广播时，ReceiverDispatcher
可以很方便地调用BreadcastReceiver的onReceive方法，和
Service的实现原理类似
4.IInterntReceiver内部有ReceiverDispatcher的引用
ActivityManager
Nativ.getDefault(
).registereceiver()
采用跨进程的方式向AMS发送广播注册的请求
真正实现注册广播，在AMS中
IntentResolver<B
roadcastFilter,
BroadcastFilter>.
addFilter
最终将InnerReceiver对象和IntentFiler对象存储起来


广播的普通发送
Context.Wrapper.sendBroadcast()
ContextImpl.sendBroadcast()
ActivityManagerNative.getDefault().BroadcastIntent()
AMS发起一个异步请求用于发送广播
BroadcastIntentLocked()
Intent.addFlas(Intent.FLAG_EXCLUDE_STOPPED_PACKAGES)
Android3.1开始，系统为所有广播默认添加了FLAG_EXCLUDE_STOPPED_PACKAGES标记。防止广播无意间或不必要 的时候调起已经停止运行的应用。
处于停止状态:
1.应用安装后未运行
2.应用被手动或者其他应用强停
FLAG_EXCLUDE_STOPPED_PACKAGES
表示不包含已经停止的应用
FLAG_INCLUDE_STOPPED_PACKAGES
表示包含已经停止的应用
如果两个都存在，以FLAG_INCLUDE_STOPPED_PACKAGES为准
将满足条件的广播接收者添加到BroadcastQueue中
根据intent-filter查找出匹配的广播接收者并经过一系列的条件过滤，最终会将满足条件的广播接收者添加到BroadcastQueue中。
ScheduleBroadcastLocked()
BroadcastQueue将广播发送给相应的广播接收者


mHandler.sendMessage(mHandler.obtainMessage(BROADCAST_INTENT_MSG,this))
没有立即发送广播，而是发送了BROADCAST_INTENT_MSG消息
BroadCastQueue收到消息后会调用processNextBroadcast方法
无序广播存储在mParallelBroadcasts中，系统会遍历mParallelBroadCasts并将其中的广播发送给他们所有的接收者。
遍历mParallelBroadcasts
无序广播存储在mParallelBroadcasts中
deliverToRegisteredReceiverLocked()
PerformReceiveLocked()
App.thread.scheduleRegisteredReceiver
App.thread仍然指ApplicationThread
LoadedApk.ReceiverDispatcher.performReceive()
mActivityThread.post(args)
Args实现了Runnable接口，mActivityThread是一个Handler，它其实就是ActivityThread中的mH，mH的类型是ActivityThread的内部类H
Args的run方法
Receiver.onReceive