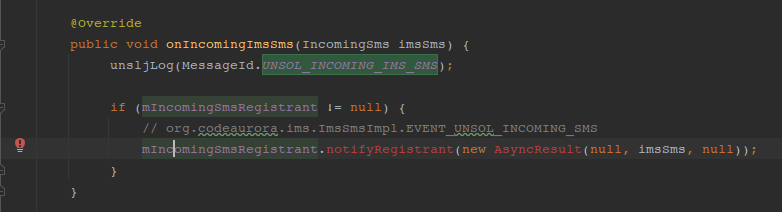
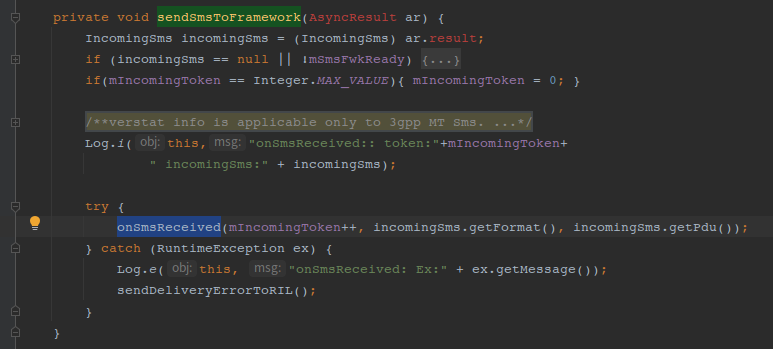
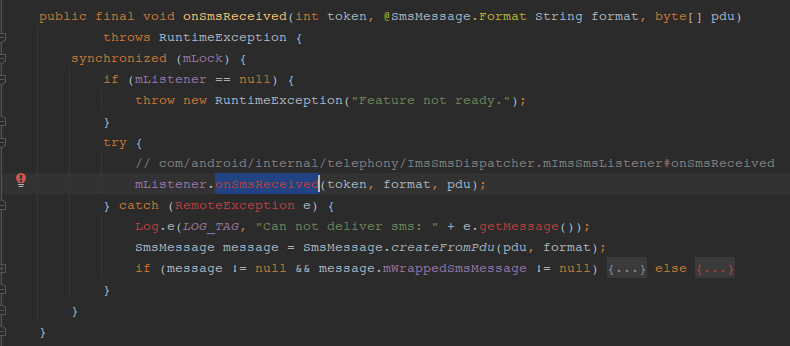
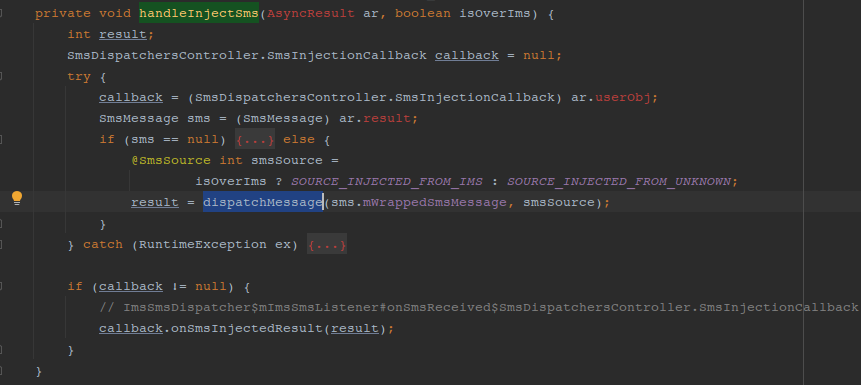
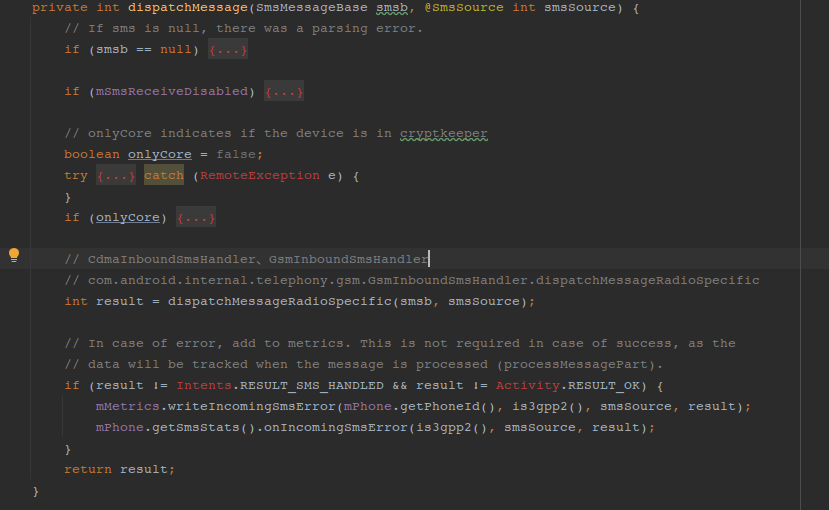
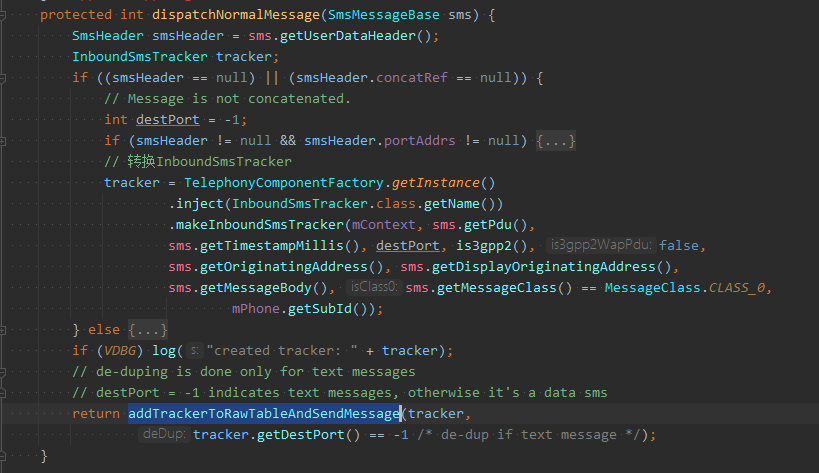
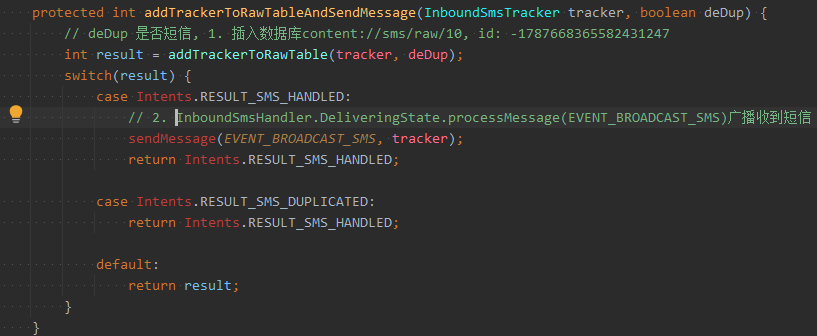
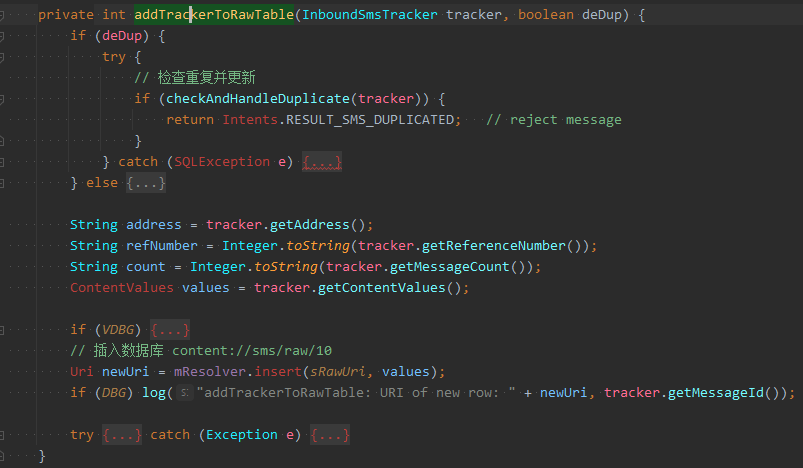
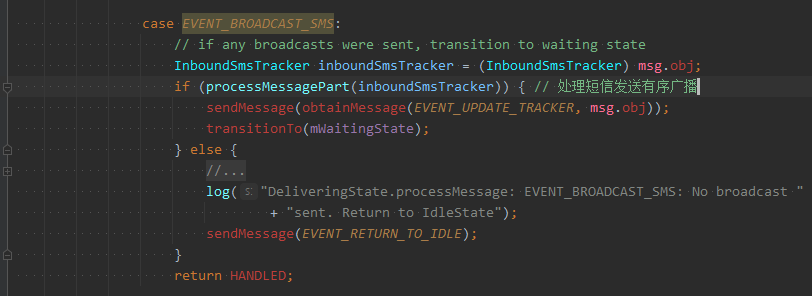
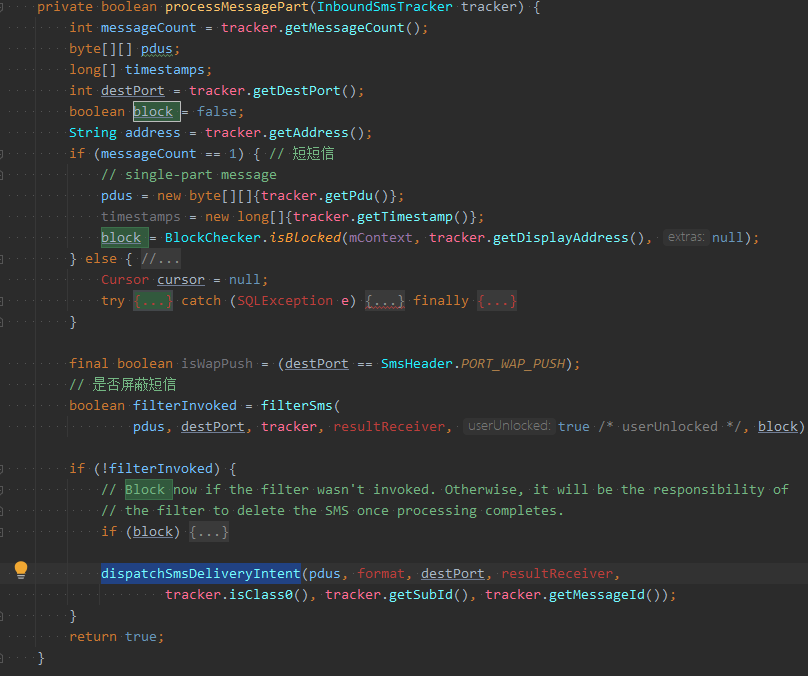
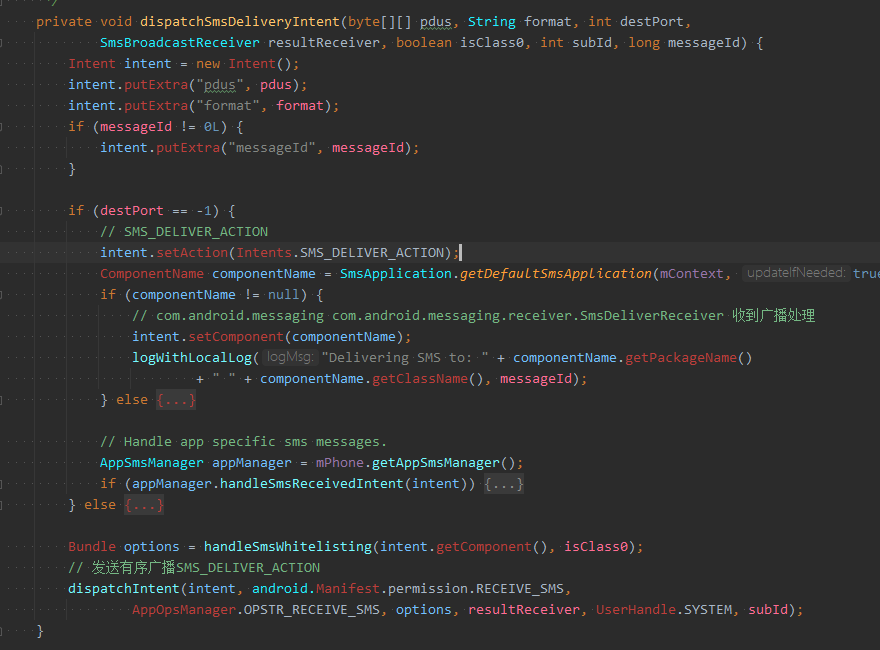
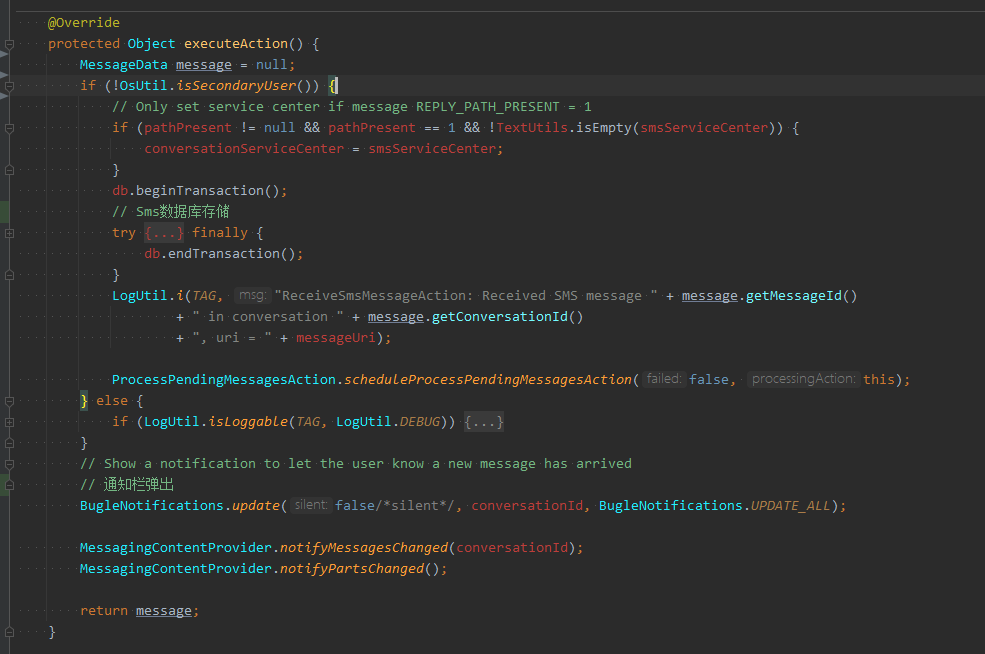
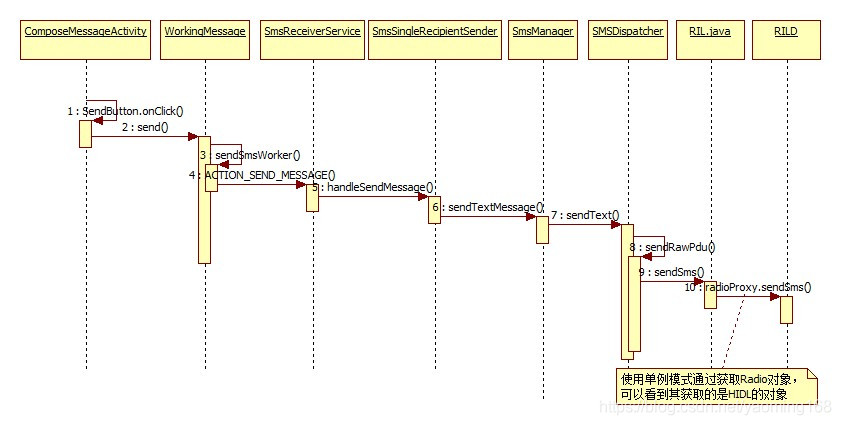
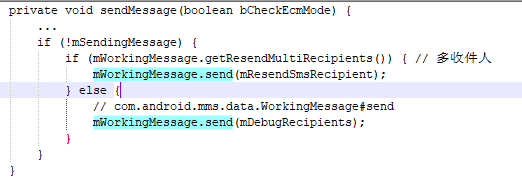
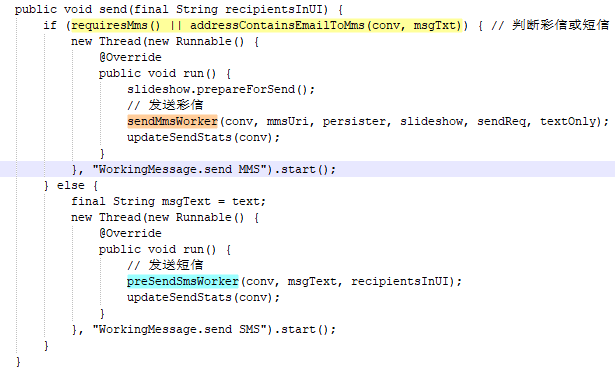
**SMS接收流程**

1. org.codeaurora.ims.ImsSenderRxr.ImsRadioIndication#onIncomingImsSms
2. org.codeaurora.ims.ImsSmsImpl#sendSmsToFramework
3. android.telephony.ims.stub.ImsSmsImplBase#onSmsReceived
4. com/android/internal/telephony/ImsSmsDispatcher.mImsSmsListener#onSmsReceived
5. com.android.internal.telephony.InboundSmsHandler.DeliveringState#processMessage - EVENT\_INJECT\_SMS → com.android.internal.telephony.InboundSmsHandler#handleInjectSms
6. com.android.internal.telephony.InboundSmsHandler#dispatchMessage(CS域短信：com.android.internal.telephony.RadioIndication#newSms → com.android.internal.telephony.InboundSmsHandler.IdleState#processMessage(EVENT\_NEW\_SMS) → com.android.internal.telephony.InboundSmsHandler.DeliveringState#processMessage(EVENT\_NEW\_SMS) → com.android.internal.telephony.InboundSmsHandler#dispatchMessage)
7. com.android.internal.telephony.gsm.GsmInboundSmsHandler.dispatchMessageRadioSpecific → com.android.internal.telephony.InboundSmsHandler#dispatchNormalMessage 
8. com.android.internal.telephony.InboundSmsHandler#addTrackerToRawTableAndSendMessage 
9. com.android.internal.telephony.InboundSmsHandler#addTrackerToRawTable 
10. com.android.internal.telephony.InboundSmsHandler.DeliveringState#processMessage(EVENT\_BROADCAST\_SMS) 
11. com.android.internal.telephony.InboundSmsHandler#processMessagePart 
12. com.android.internal.telephony.InboundSmsHandler#dispatchSmsDeliveryIntent 
13. com.android.messaging.receiver.SmsDeliverReceiver#onReceive → com.android.messaging.receiver.SmsReceiver#deliverSmsIntent → com.android.messaging.receiver.SmsReceiver#deliverSmsMessages 
14. com.android.messaging.datamodel.action.ReceiveSmsMessageAction#executeAction 

**SMS 发送流程**

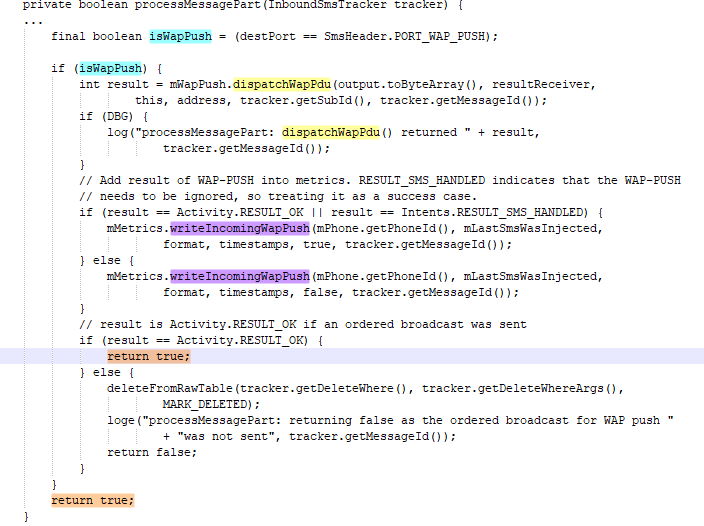
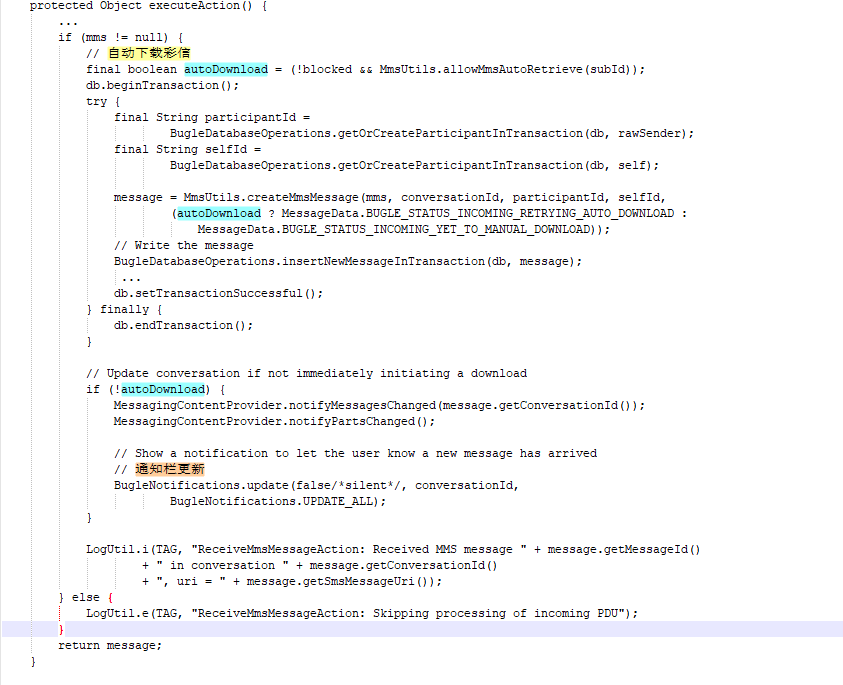


1. com.android.mms.ui.ComposeMessageActivity#sendMessage
2. com.android.mms.data.WorkingMessage#send
3. com.android.mms.data.WorkingMessage#preSendSmsWorker → com.android.mms.data.WorkingMessage#sendSmsWorker → com.android.mms.transaction.SmsMessageSender#sendMessage → com.android.mms.transaction.SmsMessageSender#queueMessage
4. android.telephony.SmsManager#sendTextMessageInternal() // 跨进程Phone进程处理
5. com.android.internal.telephony.SmsController#sendTextForSubscriber → com.android.internal.telephony.SmsController#sendIccText → com.android.internal.telephony.IccSmsInterfaceManager#sendText → com.android.internal.telephony.IccSmsInterfaceManager#sendTextInternal → com.android.internal.telephony.SmsDispatchersController#sendText → com.android.internal.telephony.SMSDispatcher#sendText → com.android.internal.telephony.SMSDispatcher#sendSubmitPdu → com.android.internal.telephony.SMSDispatcher#sendRawPdu → com.android.internal.telephony.SMSDispatcher#sendSms → com.android.internal.telephony.RIL#sendSMS // 发送AT命令

### MMS Receive

彩信的接收过程与短信略有不同，大致的流程是Frameworks会先发出一条WAPPUSH短信，告知应用程序有一个彩信，短信中含有一些信息比如过期日期，发送者手机号码，彩信的URL等，然后应用程序自行通过HTTP取回URL所指的彩信内容。

InboundSmsHandler在收到WAPPUSH短信后，会发送一个WAP\_PUSH\_DELIVER的广播，信息应用中的PushReceiver在收到此消息并进行部分解析后后开始启动TransactionService来处理新彩信。在NotificationTransaction如果设置了自动下载，在使用SmsManager.downloadMultimediaMessage来下载彩信，此方法将最终启动MmsService来下载彩信内容。具体下载过程在MmsService.downloadMessage中，下载保存后数据库更新会促使UI刷新。

1. org.codeaurora.ims.ImsSenderRxr.ImsRadioIndication#onIncomingImsSms → org.codeaurora.ims.ImsSmsImpl#sendSmsToFramework → com.android.internal.telephony.ImsSmsDispatcher$mImsSmsListener#onSmsReceived → com.android.internal.telephony.InboundSmsHandler.DeliveringState#processMessage(EVENT\_INJECT\_SMS) → com.android.internal.telephony.InboundSmsHandler#handleInjectSms → com.android.internal.telephony.InboundSmsHandler#dispatchMessage → com.android.internal.telephony.InboundSmsHandler#dispatchNormalMessage → com.android.internal.telephony.InboundSmsHandler#addTrackerToRawTableAndSendMessage(RESULT\_SMS\_HANDLED) → com.android.internal.telephony.InboundSmsHandler.DeliveringState#processMessage(EVENT\_BROADCAST\_SMS) → com.android.internal.telephony.InboundSmsHandler#processMessagePart
2. com.android.internal.telephony.WapPushOverSms#dispatchWapPdu //发送有序广播Intents.WAP\_PUSH\_DELIVER\_ACTION
3. com.android.messaging.receiver.MmsWapPushReceiver#onReceive //接收广播
4. com.android.messaging.datamodel.action.ReceiveMmsMessageAction#executeAction

### MMS发送

发送流程

1. android.telephony.SmsManager#sendMultimediaMessage
2. android.telephony.MmsManager#sendMultimediaMessage // 跨进程
3. com.android.mms.service.MmsService#mStub#sendMessage
4. com.android.mms.service.MmsService#addSimRequest
5. com.android.mms.service.MmsService#addToRunningRequestQueueSynchronized
6. com.android.mms.service.MmsRequest#execute
7. com.android.mms.service.SendRequest#doHttp // HttpURLConnectio上传彩信数据
8. com.android.mms.service.MmsRequest#processResult //通过PendingIntent广播处理结果