# Telephony

## GET\_INTENT\_SENDER\_INTENT提示权限异常

com/android/internal/telephony/dataconnection/DcTracker.java

DctConstants.***EVENT\_DATA\_SETUP\_COMPLETE***:

### onDataSetupComplete

### onDataSetupCompleteError

### DcTracker. startAlarmForReconnect

frameworks\opt\telephony\src\java\com\android\internal\telephony\dataconnection\DcTracker.java

**private void** startAlarmForReconnect(**long** delay, ApnContext apnContext) {  
 String apnType = apnContext.getApnType();  
 Intent intent = **new** Intent(***INTENT\_RECONNECT\_ALARM*** + **"."** + apnType);  
 intent.putExtra(***INTENT\_RECONNECT\_ALARM\_EXTRA\_REASON***, apnContext.getReason());  
 intent.putExtra(***INTENT\_RECONNECT\_ALARM\_EXTRA\_TYPE***, apnType);  
 intent.addFlags(Intent.***FLAG\_RECEIVER\_FOREGROUND***);  
 *// Get current sub id.* **int** subId = SubscriptionManager.*getDefaultDataSubscriptionId*();  
 intent.putExtra(PhoneConstants.***SUBSCRIPTION\_KEY***, subId);  
  
 PendingIntent alarmIntent = PendingIntent.*getBroadcast*(**mPhone**.getContext(), 0,  
 intent, PendingIntent.***FLAG\_UPDATE\_CURRENT***);  
 apnContext.setReconnectIntent(alarmIntent);  
  
 *// Use the exact timer instead of the inexact one to provide better user experience.  
 // In some extreme cases, we saw the retry was delayed for few minutes.  
 // Note that if the stated trigger time is in the past, the alarm will be triggered  
 // immediately.* **mAlarmManager**.setExact(AlarmManager.***ELAPSED\_REALTIME\_WAKEUP***,  
 SystemClock.*elapsedRealtime*() + delay, alarmIntent);  
}

### PendingIntent getBroadcast(

**public static** PendingIntent getBroadcast(Context context, **int** requestCode,  
 Intent intent, @Flags **int** flags) {  
 **return** *getBroadcastAsUser*(context, requestCode, intent, flags,  
 **new** UserHandle(UserHandle.*myUserId*()));  
}

**public static** PendingIntent getBroadcastAsUser(Context context, **int** requestCode,  
 Intent intent, **int** flags, UserHandle userHandle) {  
 String packageName = context.getPackageName();  
 String resolvedType = intent != **null** ? intent.resolveTypeIfNeeded(  
 context.getContentResolver()) : **null**;  
 **try** {  
 intent.prepareToLeaveProcess(context);  
 IIntentSender target =  
 ActivityManagerNative.*getDefault*().getIntentSender(  
 ActivityManager.***INTENT\_SENDER\_BROADCAST***, packageName,  
 **null**, **null**, requestCode, **new** Intent[] { intent },  
 resolvedType != **null** ? **new** String[] { resolvedType } : **null**,  
 flags, **null**, userHandle.getIdentifier());  
 **return** target != **null** ? **new** PendingIntent(target) : **null**;  
 } **catch** (RemoteException e) {  
 }  
 **return null**;  
}

### AMS. getIntentForIntentSender

**public** Intent getIntentForIntentSender(IIntentSender pendingResult) {  
 enforceCallingPermission(Manifest.permission.***GET\_INTENT\_SENDER\_INTENT***,  
 **"getIntentForIntentSender()"**);

原因就是没有赋予这个权限，为啥不赋予呢？

原生bug，