**修改电话中新通话记录无法立即显示联系人姓名问题**

**问题描述**

手机联系人来电或去电，通话结束后，打开通话记录列表时，新通话记录先显示电话号码。过一会儿才显示联系人姓名，体验不佳。

**问题分析**

按照Android原生设计，通话记录数据库中，除了保存电话号码，还会缓存联系人信息。当电话显示通话记录列表时，先查询通话记录数据库，显示电话号码或者缓存的联系人姓名；再针对每条通话记录去查询联系人数据库，对比缓存的联系人信息是否一致，若不一致则更新联系人信息缓存。

表1 通话记录数据库缓存的联系人信息

|  |  |
| --- | --- |
| **字段** | **说明** |
| CACHED\_NAME | 联系人姓名 |
| CACHED\_NUMBER\_TYPE | 电话号码类型，如手机，住宅 |
| CACHED\_NUMBER\_LABEL | 自定义的电话号码类型 |
| CACHED\_LOOKUP\_URI | 联系人URI，用于打开详情界面 |
| CACHED\_MATCHED\_NUMBER | 联系人数据库中对应的电话号码 |
| CACHED\_NORMALIZED\_NUMBER | E164格式的电话号码 |
| CACHED\_PHOTO\_ID | 联系人头像ID |
| CACHED\_PHOTO\_URI | 联系人头像URI |
| CACHED\_FORMATTED\_NUMBER | 按当地使用习惯格式化的电话号码 |

通话结束后，将本次通话记录保存到数据库，此时只保存电话号码，直到这条通话记录首次显示时才去缓存联系人信息。这就导致第一次显示通话记录时，只能显示电话号码，等待查询联系人数据库再更新和显示联系人姓名。

**解决思路**

针对原生设计的这个缺陷，MTK在Android M版本引入一个补丁，每次显示通话记录列表时，做联合查询（UNION），即查询通话记录数据库，查询联系人数据库，再将查询结果进行拼接（JOIN），确保每次查询后所有联系人的通话记录都能正确匹配联系人信息。如下代码片段：

|  |
| --- |
| ***Dialer/src/com/android/dialer/calllog/CallLogQueryHandler.java#fetchCalls***  *if (DialerFeatureOptions.CALL\_LOG\_UNION\_QUERY) {*  *// change CallLog query data source to calls join data view*  *uri = TelecomUtil.getCallLogUnionQueryUri(mContext).buildUpon()*  *.appendQueryParameter(Calls.LIMIT\_PARAM\_KEY, Integer.toString(limit))*  *.build();*  *} else{*  *uri = TelecomUtil.getCallLogUri(mContext).buildUpon()*  *.appendQueryParameter(Calls.LIMIT\_PARAM\_KEY, Integer.toString(limit))*  *.build();*  *}* |

但是这种方法会带来性能问题，原生做法显示通话记录只查询一次通话记录数据库，而MTK的做法，需要查询一次通话记录数据库，一次联系人数据库，再根据两次查询结果中相同电话号码进行拼接，查询速度会比原生慢。所以Android N版本，MTK放弃了这个补丁。

既然显示通话记录时再去实时查询联系人信息会带来性能问题，就只能将缓存联系人信息的时机提前，在保存通话记录同时保存对应的联系人信息。

保存通话记录时，通话记录完整信息封装在CallerInfo类（com.android.internal.telephony.CallerInfo），这是个私有类。下图描述了CallerInfo类成员与通话记录数据库字段的对应关系。

图1 通话记录数据库字段与CallerInfo对应关系

CACHED\_NAME

CACHED\_NUMBER\_TYPE

CACHED\_NUMBER\_LABEL

CACHED\_LOOKUP\_URI

CACHED\_MATCHED\_NUMBER

CACHED\_NORMALIZED\_NUMBER

CACHED\_PHOTO\_ID

CACHED\_PHOTO\_URI

CACHED\_FORMATTED\_NUMBER

name

numberType

numberLabel

contactIdOrZero

lookupKey

normalizedNumber

根据CallerInfo成员变量和通话记录数据库字段的对应关系，可额外保存联系人信息，只需要保存联系人姓名和URI，就可以确保通话记录正常显示。如下代码片段：

|  |
| --- |
| ***Telecomm/src/com/android/server/telecom/CallLogManager.java#LogCallAsyncTask***  *final Uri uri = addCall(c);*  *final CallerInfo ci = c.callerInfo;*  *if (ci != null && ci.contactIdOrZero > 0) {*  *final ContentValues values = new ContentValues();*  *values.put(Calls.CACHED\_NAME, ci.name);*  *values.put(Calls.CACHED\_NUMBER\_TYPE, ci.numberType);*  *values.put(Calls.CACHED\_NUMBER\_LABEL, ci.numberLabel);*  *values.put(Calls.CACHED\_NORMALIZED\_NUMBER, ci.normalizedNumber);*  *Uri lookupUri = Contacts.getLookupUri(ci.contactIdOrZero, ci.lookupKey);*  *if (uri != null) {*  *values.put(Calls.CACHED\_LOOKUP\_URI, lookupUri.toString());*  *}*  *c.context.getContentResolver().update(uri, values, null, null);*  *}* |

这种修改方式有两个场景例外：一是仅适用于正常通话后保存的通话记录，对导入的通话记录无能为力；二是对于紧急呼救号码（如112,，911），默认对紧急呼救号码做特殊处理，不会主动去查询联系人数据库，导致紧急呼救的通话记录无法缓存联系人信息。考虑到这两种场景使用频率很低，已经可以满足日常使用。