Sgm info 4.0 语音设置接口文档

更新记录：

更新作者	日期	增加内容描述
刘晶	2021-01-29	修正问题：3.5 节中存储所有发音人的字段 speakerId 更正为 String 类型，与系统发音人中字段 speakerId 保持一致，且值也为同样的内容
刘晶	2021-01-29	自定义提示语（响应应答语）与自定义唤醒词保持一致，需要发起在线反黄校验，补充了该部分逻辑说明。
刘晶	2021-02-28	加入 2.2 小节，设置 App 与语音 App 交互流程；
刘晶	2021-04-22	3.4 小节，加入读取形象数据接口
刘晶	2021-04-27	3.7 节中，修正一次唤醒持续交互接口
刘晶	2021-06-03	3.7.4 节中，加入自定义唤醒词设置逻辑，可见自定义唤醒词逻辑设置部分。 3.7.6 节中，一次唤醒持续交互增加了 30s 的时间配置

背景说明

Sgm 车机系统集成了百度语音 App，需要提供相应接口给到泛亚设置开发团队，以实现用户对语音相应功能的控制，如切换发音人、切换语速及打开关闭唤醒等。

1、功能概览

如图 1 所示为语音设置全部功能，加上‘帮助’总共有 9 个功能，以下将一一根据其接口进行介绍。

图 1 语音设置

2、方案介绍

2.1 方案分析

语音 App 对外输出接口，采用 ContentProvider 方式。

设置 App 更改数据库值，语音监听值变化，实现功能控制；同理，若语音 App 更改了相应的数据库值，设置 App 也监听值变化，实时刷新设置页面。

2.2 交互流程

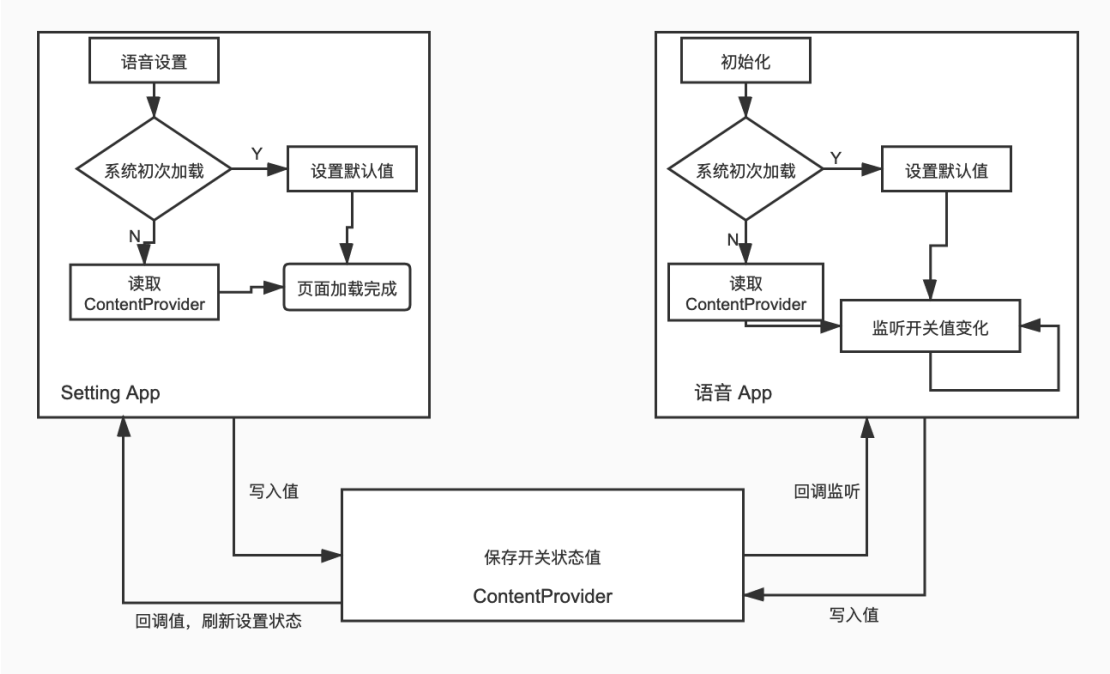


图 2 Setting App 与语音 App 交互流程

3、各功能接口

3.1 权限申请

语音设置对于外部更改设定了权限，使用前请申请权限。如下：

- **<uses-permission android:name="android.permission.READ_VR_SETTING"/>**
- **<uses-permission android:name="android.permission.WRITE_VR_SETTING"/>**

3.2 帮助

此处的需求是点击 ‘帮助’ 打开语音帮助页面，为尽量降低耦合度，此处设置 App 在点击后往 ContentProvider 中写入值，语音 App 监听值变化即可。

Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/vr_query_help_display

类型：int，每次写入数字 1 即可

示例：

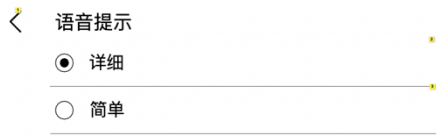
> ContentValues contentValues = new ContentValues();

> contentValues.put("setting", 1);

> mResolver.update(uri, contentValues, null, null);

3.3 语音提示

包含 2 个功能，可控制语音详细或者简洁播报；



Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/ttsSetting/tts_mode

类型：

- String，DETAIL_TTS：详尽模式（默认）；SIMPLE_TTS：简洁模式

示例：

写：

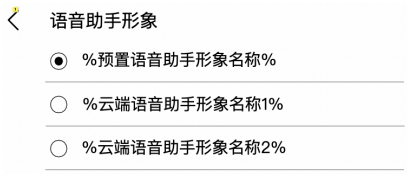
```
> ContentValues contentValues = new ContentValues();
> contentValues.put("ttsMode", "DETAIL_TTS");
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(Uri.parse(uri), null, null, null, null);
> if (cursor != null) {
>         cursor.moveToFirst();
>         String ttsMode = cursor.getString(cursor.getColumnIndex("ttsMode"));
>         cursor.close();
> }
```

3.4 语音助手形象

可控制切换语音助手形象，有>=1 个



3.4.1 语音助手形象所有数据：

该张表中，获取其中两列数据，"name"、 "comment"，name 表示形象名称，comment 标注其是否为默认形象，DEFAULT：默认形象，OTHER：非默认形象

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrAssistantInfo

类型：String

示例：

读：

```
> Cursor cursor = mResolver.query(Uri.parse(SettingItem.VR_ASSIST), null, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     do {
>         String name = cursor.getString(cursor.getColumnIndex("name"));
>         String comment = cursor.getString(cursor.getColumnIndex("comment"));
>     } while (cursor.moveToNext());
> }
> cursor.close();
```

3.4.2 语音助手选择项 Uri 接口：

将形象的选择项写入或者读取，编号从 0 开始，上面为适应 U458 彩蛋需求，默认形象会插入在表 **vrAssistantInfo** 的第一个位置，后续插入的都是其他形象。

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/vr_assistant

类型：

- Int,

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", xxx);
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     int vrAssistant = cursor.getInt(cursor.getColumnIndex("setting"));
>     cursor.close();
> }
```

3.4.3 当前语音助手名称 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrTextConfig/vr_assistant_name

类型：

- String，内部的值为当前的语音助手形象名称

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", "语音助手形象 1");
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     String vrAssistantName = cursor.getString(cursor.getColumnIndex("setting"));
>     cursor.close();
> }
```

3.5 语音发音人

3.5.1

包含 2 个大功能，控制切换系统发音人和自定义发音人

发音人切换（包含系统发音人和自定义发音人）Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/ttsSetting/tts_speaker

类型：

- String，

  系统发音人：du_xiao_wen-度小雯(默认) du_xiao_qiao-度小乔 du_xiao_lu-度小鹿 du_bo_wen-度博文 jinsha-金莎

名字对应关系如下：

-> 清纯甜美 = 度小乔

-> 活力女生 = 度小鹿

-> 情感女生 = 度小雯

-> 磁性男音 = 度博文

-> 明星音金莎 = 金莎

自定义的发音人：请写入 speakId，不要写入 speakName。

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("speakerId", "du_xiao_wen");
> mResolver.update(uri, contentValues, null, null);
```

读：

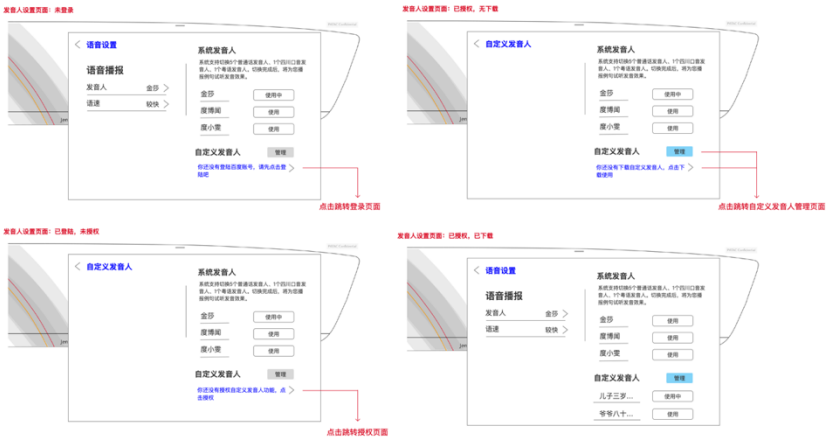
```
> Cursor cursor = mResolver.query(Uri.parse(uri), null, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     String speakerId = cursor.getString(cursor.getColumnIndex("speakerId"));
>     cursor.close();
> }
```

3.5.2 自定义发音人

自定义发音人逻辑为：打开发音人设置页面时，先判断是否登录，若未登录提示登录 Button（点击跳转百度页面）；登录之后，判断是否授权，若未授权，提示用户授权 Button（点击跳转百度页面）；授权之后，若存在自定义发音人，在设置页面展示，若不存在，提示用户下载；同时，点击页面的管理，跳转自定义发音人管理页面（百度页面）。详细也可见附件文档《TTS 发音人接口定义说明》综上分析，所有的接口信息如下：

点击下载和点击管理都进入自定义发音人管理页面；

没有登陆，没有授权，没有下载发音人，不显示管理按钮。



3.5.3 是否登录 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/ttsCustom/tts_user

类型：int，0：未登录　1：已登录

示例：这块读即可。

读：

```
> Cursor cursor = mResolver.query(uri, null, null, null, null);
> if (cursor != null && cursor.moveToFirst()) {
>     int status = cursor.getInt(cursor.getColumnIndex("status"));
>     cursor.close();
> }
```

3.5.4 是否授权 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/ttsCustom/tts_author

类型：int，0：未授权　1：已授权

示例：这块读即可。

```
> Cursor cursor = mResolver.query(uri, null, null, null, null);
> if (cursor != null && cursor.moveToFirst()) {
>     int status = cursor.getInt(cursor.getColumnIndex("status"));
>     cursor.close();
> }
```

3.5.5 如果存在自定义发音人，获取所有发音人（包括系统发音人）数据 Uri 接口如下：

- content://com.baidu.che.codriver.setting.VrSettingProvider/ttsList/tts_list

字段：如下表所示

字段名	speakerId	speakerName	speakerType	downloadStatus
类型	String	String	int	int

含义	自定义发音人 ID	自定义发音人名字	1:系统；2:自定义	1:已下载；2:未下载
示例	du_xiao_wen	度小雯		

可通过 speakerType 来区分，获取已有的自定义发音人并展示。

读：

```
> Cursor cursor = mResolver.query(Uri.parse(uri), null, null, null, null);
> if (cursor == null) {
>     return;
> }
> while (cursor.moveToNext()) {
>     int speakerId = cursor.getInt(cursor.getColumnIndex("speakerId"));
>     String speakerName = cursor.getString(cursor.getColumnIndex("speakerName"));
>     int speakerType = cursor.getInt(cursor.getColumnIndex("speakerType"));
>     int downloadStatus = cursor.getInt(cursor.getColumnIndex("downloadStatus"));
> }
> cursor.close();
```

3.5.6 跳转登录页面接口：

包名：com.baidu.che.codriver

类名：com.baidu.che.codriver.ui.CustomTtsSpeakerActivity

打开：

```
> Intent intent = new Intent();
> String packageName = "com.baidu.che.codriver";
> String className = "com.baidu.che.codriver.ui.CustomTtsSpeakerActivity";
> intent.setClassName(packageName, className);
> startActivity(intent);
```

3.5.7 跳转授权页面接口：

包名：com.baidu.che.codriver

类名：com.baidu.che.codriver.ui.CustomTtsSpeakerActivity

打开：

```
> Intent intent = new Intent();
> String packageName = "com.baidu.che.codriver";
> String className = "com.baidu.che.codriver.ui.CustomTtsSpeakerActivity";
> intent.setClassName(packageName, className);
> startActivity(intent);
```

3.5.8 跳转自定义发音人管理页面接口（与授权页面保持一致）：

包名：com.baidu.che.codriver

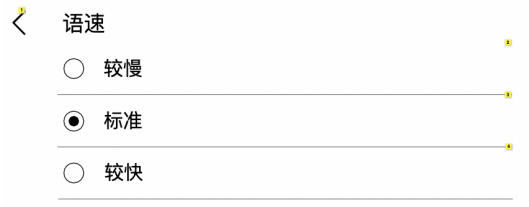
类名：com.baidu.che.codriver.ui.CustomTtsSpeakerActivity

打开：

```
> Intent intent = new Intent();
> String packageName = "com.baidu.che.codriver";
> String className = "com.baidu.che.codriver.ui.CustomTtsSpeakerActivity";
> intent.setClassName(packageName, className);
> startActivity(intent);
```

3.6 语速

可控制切换 3 种语速，较慢-标准-较快，默认为正常。



Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/ttsSetting/tts_speed

类型：

- Int，0：较慢；1：标准；2:较快

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("speed", 0);
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(Uri.parse(uri), null, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     float speedLevel = cursor.getFloat(cursor.getColumnIndex("speed"));
>     cursor.close();
}
```

3.7 唤醒设置

包含语音唤醒开关、自定义唤醒词设置、一次唤醒持续交互设置、指定说法免唤醒开关。

3.7.1 语音唤醒开关 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/wakeup_config

类型：Int，1：打开；-1：关闭

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", 1);
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     int wakeUpConfig = cursor.getInt(cursor.getColumnIndex("setting"));
>     cursor.close();
> }
```

3.7.2one-shot 开关 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/oneshot_config

类型：Int，1：打开；-1：关闭

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", 1);
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
```

```
> if (cursor != null) {
>       cursor.moveToFirst();
>       int oneShotConfig = cursor.getInt(cursor.getColumnIndex("setting"));
>       cursor.close();
> }
```

3.7.3 指定说法免唤醒开关 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/free_wakeup_word_config

类型：Int，1：打开；-1：关闭

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", 1);
mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>       cursor.moveToFirst();
>       int freeWdConfig= cursor.getInt(cursor.getColumnIndex("setting"));
>       cursor.close();
> }
```

3.7.4 自定义唤醒词配置

设置的自定义唤醒词需要向云端发起反黄校验，根据校验情况再确定是否设置成功。所以此处定义了三个 URL 接口： aks_wakeup_word、aks_wakeup_word_pre、aks_wakeup_word_result：

URL 值如下：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrTextConfig/aks_wakeup_word
- content://com.baidu.che.codriver.setting.VrSettingProvider/vrTextConfig/aks_wakeup_word_pre
- content://com.baidu.che.codriver.setting.VrSettingProvider/vrTextConfig/aks_wakeup_word_result

aks_wakeup_word_pre：唤醒应答语中间值，设置的中间值先写入该处，语音监听该字段，然后发起云端校验，校验结果写入字段 **prompt_word_result**。

aks_wakeup_word_result：校验结果中间值，json 格式，格式如下：

```
{
    "result": 1,                // 校验结果：1-成功，0-失败
```

" aksWord ": "小白", // 设置成功得到的自定义应答语

"errorMsg": "xxxx" // 如果失败，失败的原因

}

aks_wakeup_word：设置成功得到的自定义应答语，成功后，语音 app 会将值写入该处。

操作流程：向 aks_wakeup_word_pre 写入中间值，监听 aks_wakeup_word_result 的结果返回，结果如上。可根据情况显示设置的唤醒词，或提示错误原因。设置成功后，语音助手也会将正确的唤醒词写入 **aks_wakeup_word_result** 中。

注意：自定义唤醒词无默认值，也即默认为空。

3.7.5 一次唤醒持续交互开关 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/multiInteraction_config

类型：Int，1：打开；-1：关闭；

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", 1);
mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     int multiInteractionConfig= cursor.getInt(cursor.getColumnIndex("setting"));
>     cursor.close();
> }
```

3.7.6 一次唤醒持续交互配置 Uri 接口如下：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/multiInteraction_time_config

类型：int，0:3s 1:15s 2:60s 3:30s　//30s 是后期添加的，为保障兼容性，这里定为 3

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", 0);
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
```

> if (cursor != null) {

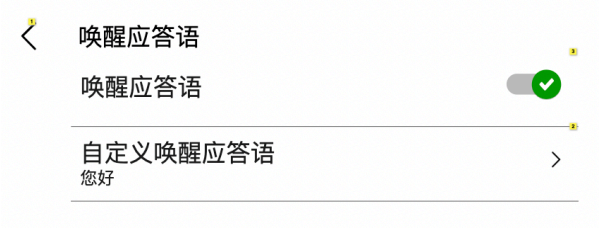
> cursor.moveToFirst();

> int multiInteractionTimeConfig= cursor.getInt(cursor.getColumnIndex("setting"));

> cursor.close();

> }

3.8 响应应答语



~~响应应答语开关 Uri 接口：~~ **开关已去除，需求已定，UE 后期更新。**

- ~~content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/prompt_enable~~

~~**类型：**int，1：开启（默认），2：关闭。~~

~~**示例：**~~

~~**写：**~~

~~> ContentValues contentValues = new ContentValues();~~

~~> contentValues.put("setting", 1);~~

~~> mResolver.update(uri, contentValues, null, null);~~

~~**读：**~~

~~> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);~~

~~> if (cursor != null) {~~

~~> cursor.moveToFirst();~~

~~> int promptEnable = cursor.getInt(cursor.getColumnIndex("setting"));~~

~~> cursor.close();~~

~~> }~~

3.8.1 响应应答语模式 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/prompt_mode

类型：int，1:响应应答提示语；2:提示音；3:自定义响应应答提示语

写：

> ContentValues contentValues = new ContentValues();

> contentValues.put("setting", 1);

> mResolver.update(uri, contentValues, null, null);

读：

> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);

> if (cursor != null) {

> cursor.moveToFirst();

> int promptEnable = cursor.getInt(cursor.getColumnIndex("setting"));

> cursor.close();

> }

3.8.2 自定义响应应答语配置

自定义响应应答语与自定义唤醒词逻辑保持一致，设置的应答语需要向云端发起反黄校验，根据校验情况再确定是否设置成功。所以此处定义了三个字段： prompt_word、prompt_word_pre、prompt_word_result：

prompt_word_pre：唤醒应答语中间值，设置的中间值先写入该处，语音监听该字段，然后发起云端校验，校验结果写入字段 **prompt_word_result**。

prompt_word_result：校验结果中间值，json 格式，格式如下：

```
{
    "result": 1,                // 校验结果：1-成功，0-失败
    "promptWord": "小白", // 设置成功得到的自定义应答语
    "errorMsg": "xxxx"          // 如果失败，失败的原因
}
```

prompt_word：设置成功得到的自定义应答语，成功后，语音 app 会将值写入该处。

注意：自定义提示语无默认值，也即默认为空。

Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrTextConfig/prompt_word

类型：String，直接写入值即可

示例：

写：

> ContentValues contentValues = new ContentValues();

> contentValues.put("setting", "xxx");

> mResolver.update(uri, contentValues, null, null);

读：

> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);

> if (cursor != null) {

> cursor.moveToFirst();

> String promptWord = cursor.getString(cursor.getColumnIndex("setting"));

> cursor.close();

> }

Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrTextConfig/prompt_word_pre

类型：String

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", "xxx");
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     String promptWord = cursor.getString(cursor.getColumnIndex("setting"));
>     cursor.close();
}
```

Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrTextConfig/prompt_word_result

类型：String

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", "xxx");
> mResolver.update(uri, contentValues, null, null);
```

读：

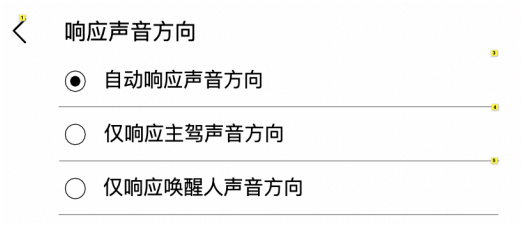
```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     String promptWord = cursor.getString(cursor.getColumnIndex("setting"));
```

> cursor.close();

}

3.9 响应声音方向

可响应 3 种声音方向。



响应声音方向 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/response_sound_direction_config

类型：int，　1-仅响应主驾声音方向 2-仅响应唤醒人声音方向（**自动响应声音方向已删除，需求已定，后期会更新 UE，可见 PIS 2030 0.5 版本**）

示例：

写：

> ContentValues contentValues = new ContentValues();

> contentValues.put("setting", 0);

> mResolver.update(uri, contentValues, null, null);

读：

> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);

> if (cursor != null) {

>　　cursor.moveToFirst();

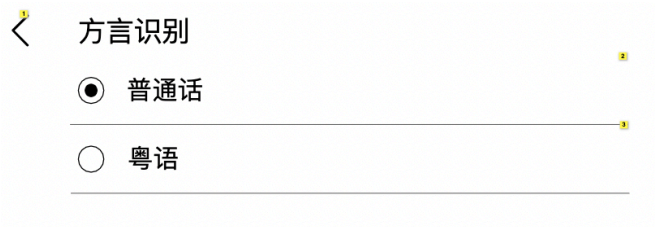
>　　int responseSoundDirectionConfig = cursor.getInt(cursor.getColumnIndex("setting"));

>　　cursor.close();

> }

3.10 方言识别

支持切换普通话和粤语。



方言识别 Uri 接口：

- content://com.baidu.che.codriver.setting.VrSettingProvider/vrSetting/vr_language_config

类型：int，0-普通话（默认）　1-粤语

示例：

写：

```
> ContentValues contentValues = new ContentValues();
> contentValues.put("setting", 0);
> mResolver.update(uri, contentValues, null, null);
```

读：

```
> Cursor cursor = mResolver.query(uri, new String[]{"id", "setting", "comment"}, null, null, null);
> if (cursor != null) {
>     cursor.moveToFirst();
>     int vr_language_config = cursor.getInt(cursor.getColumnIndex("setting"));
>     cursor.close();
> }
```

附录 1 Bridge-SDK 集成引入方式 如不需要集成该 SDK，可忽略

1、依赖 sdk

1.1 本地依赖 arr 包

取出附件中的 dueros-bridge-x.x.x.aar，放到模块的 libs 文件目录下，并在模块的 build.gradle 文件中加入如下依赖：

```
android {
    ……
    repositories {
        flatDir {
            dirs 'libs'
        }
    }
    ……
}
dependencies {
    ……
    api(name: 'dueros-bridge-x.x.x', ext: 'aar')
    ……
}
```

1.2 线上依赖 aar

aar 包已传入泛亚仓库，可直接依赖，dependencies 保持不变。

https://info-maven.apps.saic-gm.com/repository/PATAC_info4_HMI/

2、Bridge-Sdk 接入说明

2.1 初始化

在 App 中 Application 类中的 onCreate()方法中初始化 BridgeManager，建立与 DuerOS 的通信。

例如：

```
public class CodriverApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        BridgeManager.getInstance().init(this); // 初始化 Bridge-sdk，并发起绑定服务。
    }
}
```

2.2 注入回调监听

```
public interface IDirectReceiver {
    /**
```

```
     * 重点关注该函数，数据从该处回调！
     * @param pakeName 包名，不需要关注
     * @param msgType   返回消息类型，不需要关注
     * @param data          识别的文字内容
     */
    @Override
    public void onReceive(String pakeName, String msgType, final String data) {
        Log.d(TAG, "接收到的内容" + data);
        // data 即为从语音 app 发送过来的内容
    }

    /**
    * @param isConnected true，服务已经连接，可正常收发消息；fasle，不可用。
    */
    @Override
    public void onConnectionStateChanged(boolean isConnected) {
        if(isConnected) {
            Log.d(TAG, "onConnectionStateChanged: 连接成功");  // 注册监听之后，会回调该函数，判断当前 app 与 DuerOS 的连接状态
        } else {
            Log.d(TAG, "onConnectionStateChanged: 未连接");
        }
    }
}
BridgeManager.getInstance().registerDirectReceiver(this);  // 注册识别内容回调监听
```

3、发送 DuerOSEvent 至语音 app

3.1 发送内容

举例如下：

```
BridgeManager.DcsType dcsType = BridgeManager.DcsType.Event;
String data = {
  "header": {
    "dialogRequestId": "",
    "messageId": "",
    "name": "CreateReminder",
    "namespace": "ai.dueros.device_interface.extensions.iov_alert"
  },
  "payload": {}
};
BridgeManager.getInstance().send(dcsType, data);  // 发送数据至语音 app
```

这里，数据类型选择 BridgeManager.DcsType.Event 即可。

3.2 字段介绍

- **namespace**：代表语音 app 中某一模块，**固定不变**。
- **name**：代表**意图**，例如以上 CreateReminder 表示 Reminder 通知语音 app 创建日程。
- **dialogRequestId**：未使用，无需关注。
- **messageId**：未使用，无需关注。
- **payload** :这里一般装载数据内容, 如创建日程的相关数据就放入该字段内送入 Reminder。