

Android 蓝牙电话接口介绍

修改记录

版本号	日期	修改人	修改记录
1.0	2026/06/23	隋全超	整理蓝牙电话等基本操作的接口信息

概要

依照客户需求，整理 android 蓝牙电话连接，接听，挂断，拒接，切源以及电话本下载等基本操作的接口信息。

API 接口介绍

android.bluetooth.BluetoothHeadsetClient.connect

```
public boolean connect(BluetoothDevice device)
```

- 功能描述**
车机发起HFP协议连接。
- 参数介绍**
BluetoothDevice，车机要连接的设备。
- 返回值**
true：执行成功； false：执行失败。

android.bluetooth.BluetoothHeadsetClient.disconnect

```
public boolean disconnect(BluetoothDevice device)
```

- 功能描述**
车机断开HFP协议的连接。
- 参数介绍**
BluetoothDevice，车机要断开的设备。
- 返回值**
true：执行成功； false：执行失败。

android.bluetooth.BluetoothHeadsetClient.getConnectedDevices

```
public @NonNull List<BluetoothDevice> getConnectedDevices()
```

- **功能描述**

获取当前已连接的HFP设备。

- **返回值**

当前连接的HFP远端设备。一般只有一个。根据需求，也可以有多个。

android.bluetooth.BluetoothHeadsetClient.getConnectionState

```
public @BtProfileState int getConnectionState(@NonNull BluetoothDevice device)
```

- **功能描述**

获取车机与指定远端设备的HFP连接状态。

- **参数介绍**

BluetoothDevice, 指定的远端设备。

- **返回值**

连接状态。

返回的连接状态
BluetoothProfile.STATE_DISCONNECTED
BluetoothProfile.STATE_CONNECTING
BluetoothProfile.STATE_CONNECTED
BluetoothProfile.STATE_DISCONNECTING

BluetoothHeadset.ACTION_CONNECTION_STATE_CHANGED 广播

```
package com.android.bluetooth.hfpclient;

private void broadcastConnectionState(BluetoothDevice device, int newState, int
prevState) {
    logD("Connection state " + device + ": " + prevState + "->" + newState);
    /*
     * Notifying the connection state change of the profile before sending
     * the intent for connection state change, as it was causing a race
     */
}
```

```

        * condition, with the UI not being updated with the correct connection
        * state.
        */
        Intent intent = new
Intent(BluetoothHeadsetClient.ACTION_CONNECTION_STATE_CHANGED);
        intent.putExtra(BluetoothProfile.EXTRA_PREVIOUS_STATE, prevState);
        intent.putExtra(BluetoothProfile.EXTRA_STATE, newState);
        intent.putExtra(BluetoothDevice.EXTRA_DEVICE, device);

        // add feature extras when connected
        if (newState == BluetoothProfile.STATE_CONNECTED) {
            //skip ...
        }
        mService.sendBroadcastMultiplePermissions(intent,
            new String[] {BLUETOOTH_CONNECT, BLUETOOTH_PRIVILEGED},
            Utils.getTempBroadcastOptions());

        HfpClientConnectionService.onConnectionStateChanged(device, newState,
prevState);
    }

```

- **功能描述**

当 HFP 连接状态改变时，发送该广播通知车机上层。

- **参数介绍**

广播参数	参数类型	说明
BluetoothProfile.EXTRA_PREVIOUS_STATE	int	改变前的状态
BluetoothProfile.EXTRA_STATE	int	改变后的状态
BluetoothDevice.EXTRA_DEVICE	BluetoothDevice	状态改变对应的远端设备
...		

android.bluetooth.BluetoothHeadsetClient.dial

```

public BluetoothHeadsetClientCall dial(BluetoothDevice device, String number)

```

- **功能描述**

HFP连接后，在车机端拨打电话。

- **参数介绍**

参数类型	参数名称	参数说明
BluetoothDevice	device	对应的远端设备
String	number	要拨打的电话号码

- **返回值**

BluetoothHeadsetClientCall 拨打的电话信息，包含电话号码，电话状态，以及对应的远端设备等。

android.bluetooth.BluetoothHeadsetClient.acceptCall

```
public boolean acceptCall(BluetoothDevice device, int flag)
```

- 功能描述

当来电时，在车机端接听电话

- 参数介绍

参数类型	参数名称	参数说明
BluetoothDevice	device	对应的远端设备
int	flag	CALL_ACCEPT_NONE：接听来电 CALL_ACCEPT_HOLD：三方通话时，将active call 变为hold，并接听来电 CALL_ACCEPT_TERMINATE：三方通话时，将active call 挂断，并接听来电

- 返回值

true：执行成功；false：执行失败。

android.bluetooth.BluetoothHeadsetClient.terminateCall

```
public boolean terminateCall(BluetoothDevice device, BluetoothHeadsetClientCall call)
```

- 功能描述

在车机端挂断电话。

- 参数介绍

参数类型	参数名称	参数说明
BluetoothDevice	device	对应的远端设备
BluetoothHeadsetClientCall	call	要挂断的电话

- 返回值

true：执行成功；false：执行失败。

android.bluetooth.BluetoothHeadsetClient.rejectCall

```
public boolean rejectCall(BluetoothDevice device)
```

- **功能描述**
当来电时，在车机端拒接电话。
- **参数介绍**
BluetoothDevice，对应的远端设备。
- **返回值**
true：执行成功；false：执行失败。

android.bluetooth.BluetoothHeadsetClient.getCurrentCalls

```
public List<BluetoothHeadsetClientCall> getCurrentCalls(BluetoothDevice device)
```

- **功能描述**
获取当前的电话信息。
- **参数介绍**
BluetoothDevice，对应的远端设备。
- **返回值**
返回当前通话，可能不止一通。

BluetoothHeadsetClient.ACTION_CALL_CHANGED 广播

```
package com.android.bluetooth.hfpclient;

private void sendCallChangedIntent(HfpClientCall c) {
    LogD("sendCallChangedIntent " + c);
    Intent intent = new Intent(BluetoothHeadsetClient.ACTION_CALL_CHANGED);
    intent.addFlags(Intent.FLAG_RECEIVER_FOREGROUND);
    intent.putExtra(BluetoothHeadsetClient.EXTRA_CALL, c);
    Utils.sendBroadcast(mService, intent, BLUETOOTH_CONNECT,
        Utils.getTempAllowlistBroadcastOptions());
    HfpClientConnectionService.onCallChanged(c.getDevice(), c);
}
```

- **功能描述**
当通话状态改变时，发送该广播通知车机上层。
- **参数介绍**

广播参数	参数类型	参数说明
Intent.FLAG_RECEIVER_FOREGROUND	无	广播标志，前台广播

广播参数	参数类型	参数说明
BluetoothHeadsetClient.EXTRA_CALL	HfpClientCall	通话信息

android.bluetooth.BluetoothHeadsetClient.connectAudio

```
public boolean connectAudio(BluetoothDevice device)
```

- **功能描述**

建立SCO，将通话声音切到车机。

- **参数介绍**

BluetoothDevice，对应的远端设备。

- **返回值**

执行成功返回true，失败返回false。

android.bluetooth.BluetoothHeadsetClient.disconnectAudio

```
public boolean disconnectAudio(BluetoothDevice device)
```

- **功能描述**

断开SCO，将通话声音切到手机。

- **参数介绍**

BluetoothDevice，对应的远端设备。

- **返回值**

执行成功返回true，失败返回false。

android.bluetooth.BluetoothHeadsetClient.getAudioState

```
public int getAudioState(BluetoothDevice device)
```

- **功能描述**

获取HFP音源状态，即是否有建立SCO。

- **参数介绍**

BluetoothDevice，对应的远端设备。

- **返回值**

返回值	
STATE_AUDIO_DISCONNECTED	断开
STATE_AUDIO_CONNECTING	连接中

返回值	
STATE_AUDIO_CONNECTED	已连接

BluetoothHeadsetClient.ACTION_AUDIO_STATE_CHANGED 广播

```
package com.android.bluetooth.hfpclient;

void broadcastAudioState(BluetoothDevice device, int newState, int prevState) {

    BluetoothStatsLog.write(BluetoothStatsLog.BLUETOOTH_SCO_CONNECTION_STATE_CHANGED
    ,
        AdapterService.getAdapterService().obfuscateAddress(device),
        getConnectionStateFromAudioState(newState), mAudiowbs
            ? BluetoothHfpProtoEnums.SCO_CODEC_MSBC
            : BluetoothHfpProtoEnums.SCO_CODEC_CVSD,
        AdapterService.getAdapterService().getMetricId(device));
    Intent intent = new
Intent(BluetoothHeadsetClient.ACTION_AUDIO_STATE_CHANGED);
    intent.putExtra(BluetoothProfile.EXTRA_PREVIOUS_STATE, prevState);
    intent.putExtra(BluetoothProfile.EXTRA_STATE, newState);
    if (newState == BluetoothHeadsetClient.STATE_AUDIO_CONNECTED) {
        intent.putExtra(BluetoothHeadsetClient.EXTRA_AUDIO_WBS, mAudiowbs);
    }
    intent.putExtra(BluetoothDevice.EXTRA_DEVICE, device);
    Utils.sendBroadcast(mService, intent, BLUETOOTH_CONNECT,
        Utils.getTempAllowlistBroadcastOptions());

    logD("Audio state " + device + ": " + prevState + "->" + newState);
    HfpClientConnectionService.onAudioStateChanged(device, newState, prevState);
}
```

- 功能描述
当 SCO 状态改变时，使用该广播通知车机上层。
- 广播参数

广播参数	参数类型	参数说明
BluetoothProfile.EXTRA_PREVIOUS_STATE	int	改变前的状态
BluetoothProfile.EXTRA_STATE	int	改变后的状态
BluetoothDevice.EXTRA_DEVICE	BluetoothDevice	对应的远端设备
...		