



文件名称：Airplay接收端SDK接口文档
Android平台
当前版本：V1.6

苏州必捷网络科技有限公司

修订记录

版本号	拟制/修改人	拟制/修改日期	评审人	修改内容要点
0.9	吴刚	2018		初稿
1.0	方一鸣	2018		更新SDK导入方式
1.1	吴刚	2018		修改部分接口
1.2	吴刚	2019.3.13		修改音视频数据回调接口
1.3	吴刚	2019.6.21		新增音频封面信息等接口
1.4	吴刚	2019.7.8		1 新增播放器状态查询接口 2 初始化接口返回值改为int
1.5	吴刚	2020.1.14		新增投屏会话属性协商接口
1.6	吴刚	2020.3.28		1.0.29版本，PIN码设置接口新增返回值
文档初始拟定时，可不填“评审人”以及“修改内容要点”				

目 录

修订记录.....	2
目 录.....	3
1 概述.....	4
1.1 目的.....	4
1.2 读者对象.....	4
1.3 缩略语定义.....	4
2 范围.....	4
2.1 功能.....	4
2.2 SDK框架.....	4
2.3 SDK的DEMO实现.....	5
2.4 SDK交付物.....	5
3 接口.....	5
3.1 AirplayModule类中的接口说明.....	6
3.1.1 设置客户定制的实现类接口.....	6
3.1.2 初始化接口.....	6
3.1.3 去初始化接口.....	7
3.1.4 更新PIN码接口.....	7
3.1.5 强制结束某路Airplay会话接口.....	7
3.1.6 实时修改Airplay接收端名称接口.....	7
3.2 ModuleImpltf接口说明.....	7
3.2.1 会话接入接口.....	7
3.2.2 会话结束接口.....	7
3.2.3 投屏属性协商接口.....	7
3.3 MediaChannel会话处理接口.....	8
3.3.1 设置窗口句柄.....	8
3.3.2 音频数据回吐接口.....	8
3.3.3 更新音频播放的封面信息接口.....	8
3.3.4 更新音频播放的歌曲信息接口.....	8
3.3.5 视频数据回吐接口.....	8
3.3.6 音量控制接口.....	8
3.3.7 视频旋转接口.....	8
3.3.8 视频URL播放暂停接口.....	9
3.3.9 视频URL播放继续播放接口.....	9
3.3.10 视频URL播放SEEK接口.....	9
3.3.11 获取URL播放器当前播放位置接口.....	9
3.3.12 获取URL视频总时长.....	9
3.3.13 视频URL播放器当前的状态.....	9
4 Demo说明.....	9
5 客户如何使用SDK.....	10

1 概述

1.1 目的

用于指导使用必捷 Airplay 接收端 SDK 的开发人员进行开发及测试。

1.2 读者对象

本文档适用于开发 Android 平台 Airplay 接收端的开发人员。

1.3 缩略语定义

缩写名称	英文	中文
Airplay		苹果公司制定的无线通讯技术

2 范围

2.1 功能

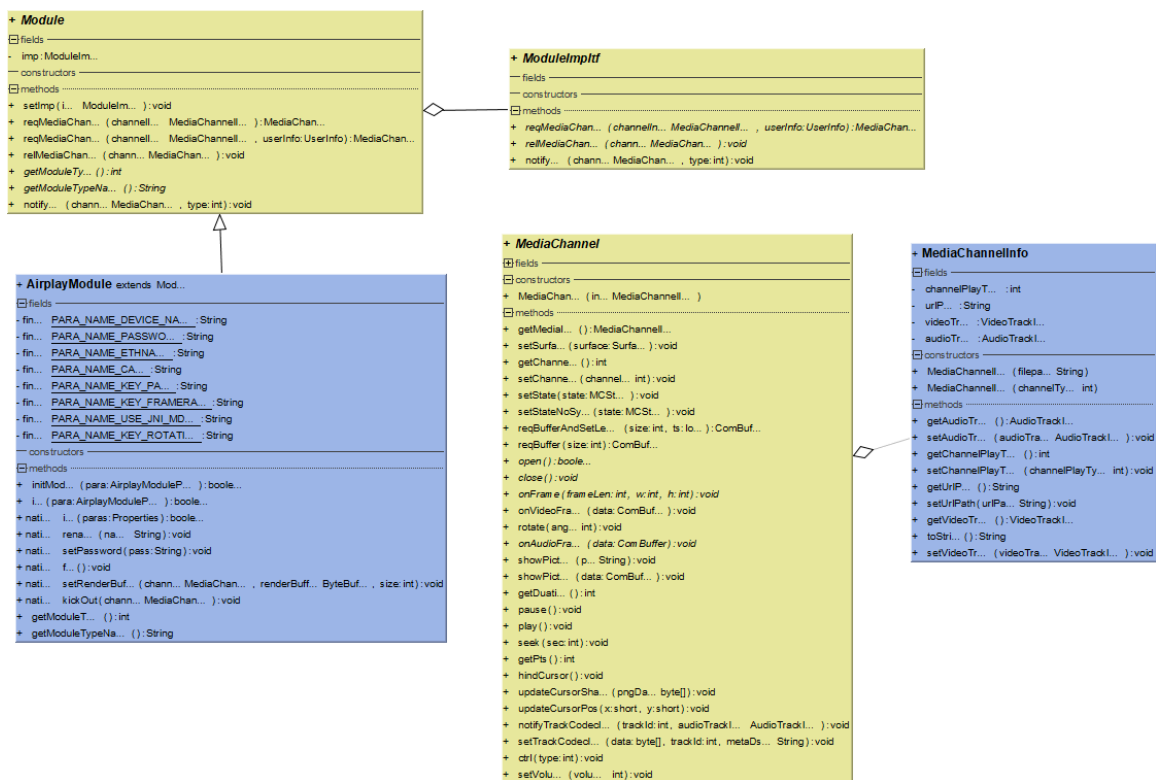
本SDK可以接受来自iphone, ipad, imac使用Airplay方式的投屏处理, 也可以支持Android手机/PC电脑使用Mirror360进行Airplay方式的投屏处理, 并提供接口给应用层调用处理。

2.2 SDK框架

BJAirplay Receiver SDK 总体分为两层

- 1) cast_base_lib-1.0.20-release.aar : 它是一个 Android Module , 它定义了基础的MediaChannel, Module 接口。
- 2) bj_airplay_lib-1.0.22-release.aar: 它是一个 Android Module, 定义了 AirplayModule, 以及相关JNI接口。

应用程序应基于 bj_airplay_lib-1.0.22-release.aar 进行开发。应用程序原则上不应修改 cast_base_lib-1.0.20-release.aar, bj_airplay_lib-1.0.22-release.aar 中的内容。



上图为核心类图，客户需要实现 ModuleImpltf 接口，和 MediaChannel 的功能接口。

2.3 SDK的DEMO实现

BJAirplayReceiver SDK Demo是接收端的一个参考实现，它基于bj_airplay_lib-1.0.17-release.aar 实现了Airplay接收端功能。其中AirplayModuleImp实现了ModuleImpltf接口，

AirplayMirrorChannel实现了MediaChannel接口，实现了对Airplay镜像会话的处理。

AirplayAudioChannel实现了MediaChannel接口，实现了对Airplay音频会话的处理。

AirplayUrlPlayChannel实现了MediaChannel接口，实现了对Airplay URL播放会话的处理。

2.4 SDK交付物

- SDK库（两个AAR文件）
- DEMO源代码
- SDK接口文档

3 接口

接口主要在AirplayModule，ModuleImpltf和MediaChannel类中进行定义；

其中AirplayModule类提供了初始化SDK，去初始化SDK，强制结束某会话等接口。

ModuleImpltf是一个接口类，需要由用户实现相关接口，SDK通过该接口类通知用户程序Airplay投屏会话的开始与结束，用户程序需要根据不同的会话类型创建对应的MediaChannel实现类和对应的播放器应用。

MediaChannel是一个接口类，需由用户程序实现，SDK通过此接口类将音视频数据，音量控制，视频旋转等控制事件通知应用程序，应用程序在对应播放器中进行处理。用户需实现MediaChannel，ModuleImpItf相关接口，也可参考我司提供的DEMO源代码实现。

3.1 AirplayModule类中的接口说明

3.1.1 设置客户定制模块实现类接口

```
public void setImp(ModuleImpItf imp)
```

输入:用户实现的ModuleImpItf接口实例。

3.1.2 初始化接口

```
public native int init(Properties props);
```

AirplayModule类的Init方法初始化Airplay接收端模块。App在启动做初始化时调用，AirplayModule对象应设计为全局只有一个。

其中Properties支持以下属性的设置：

PARA_NAME_DEVICE_NAME：接收端名称，Airplay接收端显示的名称。

PARA_NAME_PASSWORD：初始PIN码，默认空字符串，不支持pin码则输入空字符串。

PARA_NAME_KEY_ROTATION：镜像播放是否支持旋转，取值范围是0,1。默认为1

PARA_NAME_MDNS_DEAMON_ENABLE：是否启用内部MDNSD用于发布Airplay服务，取值范围是0,1。默认为1。此处建议用户填写1。

请注意：若接收端APP同时运行了BJAirplay接收端和BJCast接收端，则只需要在其中一个模块中将PARA_NAME_MDNS_DEAMON_ENABLE设为1，另一个务必设为0；若两个模块初始化时该参数都为1，会导致其中一个服务发布失败。

返回值：

```
public static final int BJ_AIRPLAY_ERROR_SUCCESS = 0,           //成功
public static final int BJ_AIRPLAY_ERROR_PORT_BINDERROR = 1;   //初始化失败，Airplay 端口绑定失败
public static final int BJ_AIRPLAY_ERROR_INIT_FAILED = 2;      //初始化失败
public static final int BJ_AIRPLAY_ERROR_INIT_RAOP_FAILED = 3; //初始化失败，raop 端口绑定失败
public static final int BJ_AIRPLAY_ERROR_VERIFY_GET_MAC_ERROR = -101;
//License 校验失败，无法获取有效的 mac 地址
public static final int BJ_AIRPLAY_ERROR_VERIFY_GET_HARDINFO_ERROR = -102;
//License 校验失败，无法获取有效的硬件信息
public static final int BJ_AIRPLAY_ERROR_VERIFY_MAC_FAILED = -103;
//License 校验失败，MAC 地址验证失败
public static final int BJ_AIRPLAY_ERROR_VERIFY_HARDINFO_FAILED = -104;
//License 校验失败，验证硬件信息失败
public static final int BJ_AIRPLAY_ERROR_VERIFY_SDKTYPE_FAILED = -105;
//License 校验失败，不是合法的 Airplay License
public static final int BJ_AIRPLAY_ERROR_VERIFY_TIME_EXPIRED = -106;
//License 校验失败，License 过期
public static final int BJ_AIRPLAY_ERROR_VERIFY_DECODELICENSE_FAILED = -107;
//License 校验失败，无效的 license key，格式错误
public static final int BJ_AIRPLAY_ERROR_VERIFY_OTHER_ERROR = -108;
//License 校验失败
public static final int BJ_AIRPLAY_ERROR_VERIFY_ERROR_EMPTY_CLICENSE = -109;
//License 校验失败，待校验的 license 不能为空
```

3.1.3 去初始化接口

```
public native void fini();
```

AirplayModule类的fini方法去初始化Airplay模块。App销毁Airplay接收端服务时调用。

3.1.4 更新PIN码接口

```
public native int setPassword(String pass);
```

更新Airplay PIN码。空字符串表示禁用PIN码，其它有效字符串表示PIN码。

输入：pass为当前PIN码。不启用pin码，此处请输入""空字符串。

返回值：0：成功，其它：失败

注意：当有Airplay投屏连接存在的情况下，不允许更新PIN码，此时调用该接口会返回错误。

3.1.5 强制结束某路Airplay会话接口

```
public native void kickOut(MediaChannel channel);
```

channel为ModuleImpl reqMediaChannel创建的对应会话。

3.1.6 实时修改Airplay接收端名称接口

```
public native void rename(String name);
```

3.2 ModuleImpl接口说明

ModuleImpl定义了一个接口类，需要用户去实现这些接口。

3.2.1 会话接入接口

```
public MediaChannel reqMediaChannel(MediaChannelInfo channelInfo,UserInfo userInfo);
```

输入：会话信息，描述其业务类型。

channelInfo描述了当前会话的类型信息。

userInfo描述了发射端的IP，设备型号，设备名称等信息。

输出：创建的 MediaChannel对象。

这是一个回调接口，当Airplay协议栈发现有会话接入时，JNI层会主动调用此接口，应用层需要实现相关逻辑，需要在客户自身的ModuleImpl实现类中去实现该逻辑。

具体实现可以参考DEMO源代码。

3.2.2 会话结束接口

```
public void relMediaChannel(MediaChannel channel)
```

回调接口，当底层收到会话结束时调用。应用层实现相关逻辑。

具体实现可参考DEMO源代码。

3.2.3 投屏属性协商接口

```
public void probeRenderAbility(Properties props);
```

回调接口，当底层发起投屏时，会调用该接口来进行参数协商。如协商当前Airplay投屏会话的最大帧率。用户可通过定制该接口来动态调整会话的最大帧率。举例：

```
@Override
public void probeRenderAbility(Properties props) {
    //用户可定制此函数，实现 Airplay 投屏会话的最大帧率，参考如下代码，将帧率设置为 60
    int framerate = 60;
    props.setProperty(AirplayModule.PARA_NAME_KEY_FRAMERATE, Integer.toString(framerate));
}
```

3.3 MediaChannel会话处理接口

当会话建立成功后，协议栈会调用MediaChannel类中的相应接口回吐数据或者获取状态。以下接口需要在MediaChannel子类中实现，客户需要根据自身实际情况实现下列接口。

MediaChannel子类中重要的接口如下,Airplay接收端需要重点关注以下接口。

3.3.1 设置窗口句柄

```
public void setSurface(Surface surface)
```

配置窗口句柄,用于render输出。Surface由用户创建的播放器视图获取。参考Demo中的实现。

3.3.2 音频数据回吐接口

```
public abstract void onAudioFrame(byte[] buffer,int len,long ts);
```

JNI回吐音频数据,音频是PCM格式的数据,用户需要实现对音频数据的播放处理功能。具体实现可以参考SDK中DEMO源代码。

其中buffer为音频数据数组, len为数据长度, ts为时间戳。

该接口涉及Airplay纯音频投屏会话, 镜像投屏会话。

3.3.3 更新音频播放的封面信息接口

```
public void onRefreshCoverArt(byte[] buffer,int len);
```

JNI回吐音乐封面信息, 是jpe格式的数据, buffer是jpg的内容, len为jpg文件内容长度。

该接口涉及Airplay纯音频投屏会话。

3.3.4 更新音频播放的歌曲信息接口

```
public void onRefreshTrackInfo(String album,String title,String artist)
```

JNI回吐音乐的标题, 艺术家, 唱片信息, title为标题, artist为艺术家信息, album为唱片信息(某些应用播放器实际传的为歌词)。

该接口涉及Airplay纯音频投屏会话。

3.3.5 视频数据回吐接口

```
public void onVideoFrame(byte[] buffer,int len,long ts)
```

镜像视频数据接口, 该接口中吐出的是H264视频数据, 用户需要进行解码并播放。

其中buffer为视频数据数组, len为数据长度, ts为时间戳。

具体实现可以参考SDK中DEMO源代码。

该接口涉及Airplay镜像投屏会话。

3.3.6 音量控制接口

```
public void setVolume(int volume)
```

volume取值范围为0-100。0表示静音, 100表示音量为最高。

该接口涉及Airplay纯音频投屏会话, 镜像投屏会话。

3.3.7 视频旋转接口

```
public void rotate(int angle)
```

该接口通知播放器需要对视频进行旋转, 取值范围为0,90,180,270。

当3.1.2的init接口初始化时PARA_NAME_KEY_ROTATION设定为1时, 该接口在Airplay镜像投屏时会被触发。

该接口涉及Airplay镜像投屏。

3.3.8 视频URL播放暂停接口

```
public void pause()
```

该接口通知播放器暂停视频播放。

该接口涉及 Airplay URL 投屏播放场景。

3.3.9 视频URL播放继续播放接口

```
public void play()
```

该接口通知播放器开始或继续视频播放。

该接口涉及 Airplay URL 投屏播放场景。

3.3.10 视频URL播放SEEK接口

```
public void seek(int sec)
```

通知播放器进度条拖放到的位置。单位为秒。

该接口涉及 Airplay URL 投屏播放场景。

3.3.11 获取URL播放器当前播放位置接口

```
public int getPts()
```

获取播放器当前播放到的位置。单位为毫秒。

该接口涉及 Airplay URL 投屏播放场景。

3.3.12 获取URL视频总时长

```
public int getDuration()
```

获取当前播放的视频URL的时长。单位为毫秒。

该接口涉及 Airplay URL 投屏播放场景。

3.3.13 视频URL播放器当前的状态

```
public int getPlayerStatus();
```

SDK内部会播放器当前的实际状态。客户需要正确实现该接口，并返回其状态。参考demo中的实现。

该接口涉及 Airplay URL 投屏播放场景。

返回值取值范围在MediaConst类中定义为：

```
public static final int PLAYER_STATUS_LOADING = 0;  
public static final int PLAYER_STATUS_PLAYING = 1;  
public static final int PLAYER_STATUS_PAUSED = 2;  
public static final int PLAYER_STATUS_ENDED = 3;  
public static final int PLAYER_STATUS_FAILED = 4;
```

其含义描述如下

LOADING:表示在加载中，初始化的时候会出现

PLAYING: 播放中

PAUSED: 暂停中

ENDED: 播放结束

FAILED: 播放失败

4 Demo说明

Demo中AirplayModuleImp实现了ModuleImplf接口，实现了reqMediaChannel,relMediaChannel方法，此处可以控制是否接入某个会话的逻辑控制(如最大接入2路这样的控制)。其中reqMediaChannel中创建了与会话类型对应的MediaChannel实现，并启动播

放相关的View，创建用于播放的Surface。 AirplayMirrorChannel, AirplayAudioChannel, AirplayUriPlayChannel分别对应airplay镜像投屏，纯音频投屏，视频URL播放这几种业务类型，实现了MediaChannel中相关功能接口。

5 客户如何使用SDK

- 1) 导入aar，在app目录下新建libs目录并将aar文件放在该目录下，然后在build.gradle（app）中dependencies上方及内部分别添加如下代码：

```
repositories { flatDir { dirs 'libs' } }
compile(name: 'bj_airplay_lib-1.0.22-release.aar', ext: 'aar')
compile(name: 'cast_base_lib-1.0.20-release.aar', ext: 'aar')
```

，可参考Demo。
- 2) 实现ModuleImplItf接口，参考AirplayModuleImp。reqMediaChannel接口返回自定义的MediaChannel实现类的实例，并启动播放界面。
- 3) 实现MediaChannel相关功能接口，参考AirplayMirrorChannel, AirplayAudioChannel, AirplayUriPlayChannel。
- 4) 实现自身的播放界面，可参考view这个package中的view的实现，将播放界面的Surface设置到MediaChannel中。