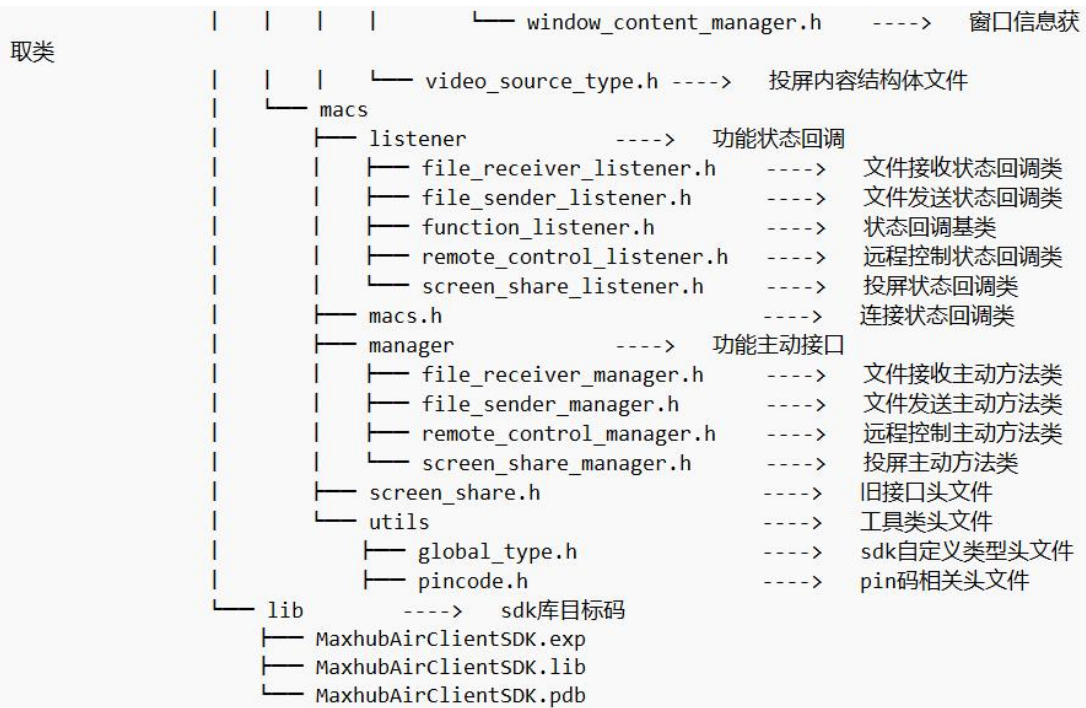


```

# 概述
本文将对PC发送端SDK进行接入说明
## sdk内容
1. 头文件
1. 库文件
1. sdk示例源代码与目标码
2. Doxygen接口帮助文档
### windows
提供dll与lib
### macos
提供framework及OC接口
# 历史版本
| SDK版本号 | 日期 | 说明 |
|-----|-----|-----|
| 2.1.10 | 2020/9/28 | 初始SDK版本 |
| 3.2.20 | 2021/5/10 | SDK接口重构版本 |
| 3.5.61 | 2021/11/1 | BYOD功能, 安全模式功能, MacOS12适配版本 |
| 3.8.114 | 2022/8/12 | 修改demo程序 |
| 3.9.90 | 2022/9/14 | 更新SDK文档 |
| 3.9.126 | 2022/10/18 | 调整AMD软编策略、修复OC接口OnDisconnect可能出现的死锁问题 |
| 3.10.60 | 2023/2/17 | 添加amd硬编、windows下使用虚拟声卡策略 |
# sdk文件夹说明
下以windows为例
```
[~/MaxhubAirClientSdk]
-> % tree -d
.
├── doc
│ ├── html ----> doxygen接口帮助文档
│ │ └── index.html ----> doxygen文档入口
│ └── README.md ----> sdk帮助文档
├── example
│ ├── cli_demo ----> 旧接口示例程序源文件
│ └── new_cli_demo ----> 新接口示例程序源文件
│ ├── file_receiver_service ----> 文件接收功能服务示例代码
│ ├── file_sender_service ----> 文件发送功能服务示例代码
│ └── macs_obj ----> 基础连接功能示例代码
│ ├── remote_control_service ----> 远程控制功能服务示例代码
│ ├── byod_client_service ----> BYOD功能服务示例代码
│ └── screen_share_service ----> 投屏功能服务示例代码
├── oc_demo ----> OC接口示例程序源文件
├── macos ----> windows sdk
├── windows ----> windows sdk
│ └── x86
│ ├── debug
│ └── release
│ └── conan_package
│ ├── bin
│ │ ├── MaxhubAirClientSDK.dll ----> windows dll文件
│ │ └── VideoSource.dll ----> windows dll文件
│ ├── include ----> sdk头文件目录
│ │ ├── third
│ │ │ ├── video_common ----> VideoSource头文件依赖
│ │ │ ├── nlohmann ----> VideoSource头文件依赖
│ │ │ ├── VideoSource ----> 屏幕和窗口信息获取接口头文件
│ │ │ └── content
│ │ │ ├── screen ----> 屏幕信息获取接口头文件
│ │ │ │ ├── screen_content.h ----> 屏幕信息类
│ │ │ │ └── screen_content_manager.h ----> 屏幕信息获取类
│ │ │ └── video_content.h ----> 投屏内容接口类
│ │ └── window ----> 窗口信息获取接口头文件
│ │ └── window_content.h ----> 窗口信息类
└── 取类
 ├── video_content.h ----> 投屏内容接口类
 ├── window ----> 窗口信息获取接口头文件
 └── window_content.h ----> 窗口信息类

```



取类

- 接入示例见example文件夹，可直接使用cmake进行构建example示例程序
- 详细接口说明参见doxygen文档
- 使用cmake生成xcode/vs工程或直接编译命令行程序

## 新接口文件接入说明

1. 包含头文件macs/mac.h并定义Macos对象
1. 继承IConnectListener，以获取连接状态
1. 通过Macos的SetListener设置连接状态回调对象
1. 按功能需求，继承listener文件夹下的对应功能的listener（例如 macs/listener/screen\_share\_listener.h），并通过Macos的AddFunction添加功能(注: AddFunction需在Connect之前调用，否则添加功能无效)

1. 使用Macos的Connect方法连接接收端

1. 通过步骤4继承的listener，回调的OnInitSuccess方法拿到manager对象，使用manager调用对应功能的主动方法

### 功能介绍

功能添加步骤见[功能添加步骤](#define\_function)

#### 投屏功能

##### 前置步骤

\*\*macos\*\*需要申请屏幕录制权限

##### api基础流程说明

Connect -> OnConnected

Start -> OnStarted

Stop -> OnStopped

Disconnected

#### 触摸回传功能

##### 前置步骤

\*\*macos\*\*需要申请辅助功能权限

\*\*windows\*\*需要使用管理员权限安装触摸驱动([驱动安装步骤](#driver\_install))

##### api基础流程说明

Connect -> OnConnected

Start -> OnStarted

Enable(true)

Stop -> OnStopped

Disconnected

```
驱动说明
```

```
windows
```

```
安装命令: bat drivers/install.cmd
```

```
```
```

```
├──drivers
│   ├──cert
│   ├──tools      ---->  驱动安装工具
│   ├──vaudio     ---->  虚拟声卡及麦克风驱动
│   ├──vcamera    ---->  虚拟摄像头驱动
│   ├──vhid       ---->  触摸回传驱动
│   └──vmonitor   ---->  虚拟拓展屏驱动
```

```
```
```

```
macos
```

```
驱动存放在framework中,运行时动态判断是否需要安装,提示用户输入密码获取root权限
```

```
OC接口接入说明
```

1. 导入框架,在\*\*target\*\*的\*\*Frameworks, Libraries, and Embedded Content\*\*中选择包含方式为\*\*Embed & Sign\*\*

2. 包含头文件

```
```objective-c
#import <MaxhubAirClientsSDK/MaxhubAirClientsSDK.h>
```
```

3. 创建client并按需注册服务

```
```objective-c
airClient = [[MHAirClient alloc] init];
NSArray *services = @[MHBODYClientService.class,

    MHFileReceiverService.class,

    MHFileSenderService.class,

    MHRemoteControlService.class,
```

```
    MHScreenShareService.class];
```

```
[airClient registerServices:services];
```
```

4. 连接对应设备

```
```objective-c
[[AirClientsSDKSingleton sharedInstance] connectWithPinCode:pincode];
```
```

5. 检查对应权限并设置,功能权限对应如下(详细检测代码可见demo的\*\*MHPermissionManager\*\*)

```
```
```

```
投屏 -> 屏幕录制
触摸回传 -> 辅助功能
```
```

6. 使用对应服务(以传屏为例)

```
```objective-c
MHScreenVideoContent *videoContent = [MHScreenVideoContent
createDefaultScreenContent];
[self.screenShareService setVideoContent:videoContent];
[self.screenShareService startCast];
```
```

## 7. 程序退出时需要先释放MHAirClient对象，防止退出死锁问题

```
`` `objective-c
AppDelegate.m

- (void)applicationWillTerminate:(NSNotification *)aNotification {
 [airClient disconnect];
 airClient = nil;
}
`` `
```

### ## OC Demo使用说明

1. 替换lib目录下的MaxhubAirClientSDK.framework文件
2. 执行pod update命令 (依赖cocoapods)
3. 运行demo, 替换签名即可

### ## 旧接口接入说明

#### ### 基础投屏

1. 包含screen\_share.h头文件，并实现IListener接口，接收结果回调
2. 定义ScreenShare实例/继承ScreenShare类
3. 使用Connect接口连接接收端
3. 使用OccupyCast (抢占投屏)、JoinCast (加入投屏)、Disconnect (断开投屏)、PauseCast (暂停投屏)、Resume (恢复投屏) 进行投屏

#### ### 投屏进阶

##### #### 应用投屏

1. 包含third/VideoSource/content/window/window\_content\_manager.h头文件
2. 通过GetAvaivableContents获取窗口id
2. 通过SetWindowShare设置应用投屏id

##### #### 拓展屏

1. 通过OpenVirtualDisplay开启、关闭拓展屏
2. SetDisplayId设置步骤1获取的screen\_id, 从而进行虚拟拓展屏投屏

...

##### ### 触摸回传

1. 通过SetTouchEnable开启、关闭触摸回传

##### ### 通用信息

###### #### pin码

1. 通过ParsePinCode解析pin码, 得到ip、port

##### # SDK日志说明

###### ## 日志路径

| 平台      | 日志默认目录                     |
|---------|----------------------------|
| Windows | ~\appdata%\<APP_NAME>\log` |
| macOS   | ~/Library/Logs/<APP_NAME>` |

###### ## 环境变量

ESLOG 支持通过环境变量进行行为控制。环境变量具有更高的优先级，会覆盖代码中的相关设置，如日志等级等。应用可以通过代码（如`putenv`）或直接修改系统环境变量的方式进行修改。需要注意的是，ESLOG仅会在初始化时读取一次环境变量，因此应用需要在ESLOG初始化前（即在调用ESLOG库提供的任何方法之前）设置好环境变量，否则会无效。

当前支持的环境变量如下

| 环境变量名 | 值类型   |
|-------|-------|
| 说明    |       |
| ----- | ----- |
| ----- | ----- |

```

| ----- |
| ESLOG_APP_NAME | 字符串
| 应用名 |
| ESLOG_LEVEL | 字符串枚举 (`trace`/`debug`/`info`/`warning`/`error`/
`critical`/`off`) | 日志输出等级
| ESLOG_SINKS | 字符串枚举 (`file`/`ide`/`console`), 通过`;`分割
| 日志Sinks, 支持同时开启多个 |
| ESLOG_FILE_DIR | 目录路径
| 日志文件保存目录 |
| ESLOG_FILE_MAX_KEEP_DAYS | 正整数
| 日志文件保留天数, 单位: 天 |
| ESLOG_FILE_MAX_MEGABYTES | 正整数
| 日志文件最大大小, 单位: MB |

常见错误说明
常见错误码说明
错误码

概览
CRCP 5.8及之前错误码见CRCP错误码

错误码

概览

错误码	说明
>0	系统错误码
-1	心跳丢失
-2	连接正常关闭
-3	收到RST, 连接被迫关闭
-4	CRCP版本不匹配
-5	认证码错误, 认证失败

-6	加密签名生成失败
-7	建立TCP连接错误
-8	损坏的消息, 消息无法解析
-9	非法的消息, 服务端无法响应此请求
-10	未知的消息ID
-11	未知错误
-12	建立TCP连接超时
-13	接收数据超时
-14	会话未找到
-15	验证签名失败
-16	不支持安全模式
-17	发送端超时未输入安全模式验证码
-18	未找到子会话对应的主会话
-19	未找到子会话编号
-20	错误的子会话密钥
-21	未找到子会话密钥
-22	本地网络故障
-23	其他网络错误
-100	连接码解析错误

详细说明

错误码>0
- 说明

投屏使用标准socket, 错误码>0请检查对应系统错误

```

- 可能原因
- 1. 系统错误

- 排查指南

1. Windows请参考<https://docs.microsoft.com/en-us/windows/win32/debug/system-error-codes>
2. Unix请参考<https://www.ibm.com/docs/en/zos/2.2.0?topic=codes-sockets-return-errno>

### ### 错误码-1

- 说明

连接方/被连接方在一定时间内（默认是4s，可通过`SetHeartBeatTimeout`来修改）未收到来自另一端的心跳则会触发该错误。

- 可能原因

1. Dongle正常使用过程中直接拔出（正常现象）
2. 另一端设备断网、断电等突然断开（正常现象）
3. 另一端心跳发送异常（常见）
4. 本端未接收心跳（常见）

- 排查指南

1. 检查是否原因1和原因2
2. 在两端开启抓包工具，检查另一方是否有发送心跳包数据（心跳包为16字节，端口可通过分析日志获得）
3. 根据抓包结果，区分是原因3或原因4，再继续分析

- 事例

1. CRCP 5.0.0 - 5.6.4之前版本中在Dongle发送端可能会因条件变量唤醒时序问题导致Dongle连接方不再接收心跳包，进一步导致误判心跳超时断开。（0426专项重点问题）
2. MaxhubAirClientSDK 1.7.15之前版本中，因macOS下存在snap机制，会导致系统将后台程序挂起，teams（现mindlinker）集成的传屏连接方SDK无法发送心跳。

- 注意

若日志中有非常明显的反复因心跳断开后2-4s后重新连接，则必定存在问题

### ### 错误码-2

- 说明

CRCP会话的TCP连接被另一端主动关闭，收到FIN包后会触发该错误，正常情况下也可能触发该错误，对应TCP socket错误码中的EOF

- 可能原因

1. 另一端在正常情况下主动断开了连接（常见）
2. 另一端发生了崩溃

- 排查指南

1. 检查另一端应用是否退出或系统关机

### ### 错误码-3

- 说明

在一端主动或异常断开TCP连接时，另一端会在收到TCP的RST包后触发该错误。

- 可能原因

1. 另一端应用崩溃（常见）

- 排查指南

1. 检查另一端应用是否发生崩溃
2. 分析日志, 检查应用逻辑

### 错误码-4

- 说明

在CRCP连接过程的握手阶段, 一端检测到另一端的CRCP版本是不兼容的版本会触发该错误。截至目前, CRCP一直是保持向后兼容的, 故该错误码尚未被使用。

- 可能原因

1. 两端CRCP版本不兼容

- 排查指南

1. 获取两端CRCP版本, 检查是否兼容

### 错误码-5

- 说明

在CRCP连接过程中的握手阶段, 连接方使用了非默认认证码, 但被连接方对该认证码认证失败。(当前认证码作为传屏码认证使用)。

- 可能原因

2. 连接方的认证码错误 (常见)

- 排查指南

1. 分析被连接方日志, 找出被连接方当前正确的认证码以及连接方实际发送过来的认证码, 并进一步分析

- 事例

1. Android发送端传屏码连接功能在开发过程中, 传至接收端的连接码使用了小写字母, 导致认证失败

### 错误码-6

- 说明

在CRCP连接过程中的签名交换步骤, 一方无法生成签名

- 可能原因

1. 该方对另一方的签名验证失败

### 错误码-7

- 说明

CRCP在建立TCP连接时失败, 目标IP和端口访问失败

- 可能原因

1. 被连接方未正常运行 (常见)
2. 连接的端口错误 (少见)
3. 被连接方CRCP未成功监听7382-7385间任一端口 (少见)
4. 连接方网络未连接

#### - 排查指南

1. 检查网络连接
2. 检查被连接方是否正常运行
3. 检查被连接方日志中，CRCP是否成功监听端口（绝大部分情况是7382）

#### - 事例

1. 二代机中，由于存在端口映射，在规则配置有误的情况下，发送端无法访问到PC模块，表现为发送端显示错误码-7

### 错误码-8

#### - 说明

CRCP收到了二进制数据，但无法正常解析出一个CRCP数据包

#### - 可能原因

1. TCP数据传输过程中发生了错误（常见）
2. 错误地连接到了一个非CRCP监听的端口，另一端发送了未知数据（少见）
2. CRCP协议更新，产生了不兼容（少见）

#### - 排查指南

1. 通过日志/netstat等确认程序端口占用是否正常
2. 根据日志中打印的包数据分析数据错误原因
3. 数据接收端抓包排查数据发送还是接收存在问题

#### - 事例

1. 有线传屏开发过程中，二进制数据包数据在通过嵌入式层转发时出错，接收端收到非法数据

### 错误码-9

#### - 说明

CRCP收到了一个无法正常进行json反序列化或缺少必须字段的消息

#### - 可能原因

1. TCP数据传输过程中发生了错误（常见）
2. CRCP协议更新，产生了不兼容（少见）

#### - 排查指南

1. 对比接收端日志和发送端日志中该条消息的内容，确定接收端数据是否生成正常
2. 数据接收端抓包排查数据发送还是接收存在问题

#### - 事例

1. 在修复macOS 10.15问题后，出现数据错乱的问题，json反序列化不正常

### 错误码-10

#### - 说明

CRCP收到了一个消息，但该消息ID是一个当前无法处理的ID

#### - 可能原因

1. CRCP协议更新，产生了不兼容



- 排查指南

1. 检查两方CRCP版本

### 错误码-11

- 说明

未知错误，当前该错误码未使用

### 错误码-12

- 说明

CRCP在建立TCP连接时失败，连接超时

- 可能原因

1. 因网络配置错误或防火墙等原因导致网络不通（常见）
2. 二维码/传屏码等连接信息有误（一般）

- 排查指南

1. 在连接方使用ping命令向被连接方IP发送PING包，检查被连接方是否可以ping通（注意排除防火墙的干扰）
2. 停止被连接方运行，在被连接方执行nc命令监听7382端口，并在连接方使用nc检查是否可以连接上被连接方并正常收发数据

### 错误码-13

- 说明

TCP已连接成功，但在CRCP会话建立过程中，连接方5s内未收到被连接方的回复或被连接方5s内未收到连接方的下一步请求。

- 可能原因

1. 被连接方CRCP因阻塞等无法及时处理连接请求，导致连接方等待超时（常见）
1. 连接过程中连接方的请求或被连接方的回复丢失

- 排查指南

1. 分析双方日志，检查是否发送正确请求或回复
2. 分析被连接方日志，检查是否包含`stuck`相关的日志，排查接收端CRCP任务线程是否被阻塞
3. 通过内存转储，调试器附加等方式获取被连接方程序运行堆栈，结合被连接方的日志，分析是否存在阻塞

### 错误码-14

- 说明

给定的会话ID不存在（该错误在Dongle连接过程中理论上不会出现，但可能在其他过程中的错误回调中）。

- 可能原因

1. 执行操作时该会话已被移除
2. 传入了错误的会话ID

- 排查指南

1. 分析双方日志，检查会话处理流程及会话ID

### 错误码-15

- 说明

CRCP会话建立过程中，一端对另一端发送过来的签名校验失败。

- 可能原因

1. 恶意攻击

### 错误码-16

- 说明

被连接方开启了安全模式，但连接方不支持安全模式，无法进行验证导致连接被拒绝

- 可能原因

1. 连接方版本较旧

- 排查指南

1. 检查被连接方是否开启安全模式
2. 关闭被连接方的安全模式，检查是否可以连接上
2. 检查连接方CRCP版本是否低于5.10

### 错误码-17

- 说明

被连接方开启了安全模式，但连接方在60秒内未输入正确的验证码，连接方主动中断了连接

- 可能原因

- 可能原因

1. 连接方未及时输入安全模式验证码

- 排查指南

1. 检查连接方日志，查看验证记录

### 错误码-18

- 说明

连接方为子会话，但无法找到该子会话对应的主会话

- 可能原因

1. 连接方未成功建立主会话
1. 连接方建立的主会话已断开

- 排查指南

1. 检查连接方日志，检查主会话是否建立

### 错误码-19

- 说明

连接方为子会话，但被连接方无法解析出正确的子会话编号

- 可能原因

1. 连接方子会话UUID格式错误

- 排查指南

1. 检查连接方子会话UUID

### 错误码-20

- 说明

连接方为子会话，但认证密钥错误

- 可能原因

1. 连接方发送了错误的认证密钥

- 排查指南

1. 检查连接方日志，查看主会话生成的认证密钥是否正确传递

### 错误码-21

- 说明

被连接方为子会话，但未携带认证密钥

- 可能原因

1. 连接方未发送认证密钥

- 排查指南

1. 检查连接方日志，查看子会话是否从主会话获取了认证密钥

### 错误码-22

- 说明

本地网络发生变动

- 可能原因

1. 网卡被禁用  
2. 切换了网络

- 排查指南

1. 检查本地网络状态

### 错误码-23

- 说明

会话的网络连接收到了其他错误码

- 可能原因

1. 网络出现未知异常

- 排查指南

1. 检查日志，查看具体网络错误码并进一步分析

### ### 错误码-100

#### - 说明

使用连接码的接收端，需要使用连接码进行连接。解析连接码需走公网，向服务器请求对应的传屏码，然后再进行投屏。

#### - 可能原因

1. 连接码输入错误
2. 无公网网络

#### - 排查指南

1. 检查连接码是否输入正确
2. 检查是否能访问公网