NetSDK 编程指导手册
（热成像产品分册）



V2.0.0

前言

目的

欢迎使用 NetSDK（以下简称 SDK）编程指导手册。

SDK 是软件开发者在开发网络硬盘录像机、网络视频服务器、网络摄像机、网络球机和智能设备等产品监控联网应用时的开发套件。

本文档描述了热成像（Thermal Camera，简称 TPC）产品的通用业务涉及的 SDK 接口以及调用流程，更多功能接口、结构体等说明请参见《网络 SDK 开发手册》，基础核心业务流程（例如初始化、登录、普通报警和智能报警等）详见《NetSDK 编程指导手册》。

本文档提供的示例代码仅为演示接口调用方法，不保证能直接拷贝编译。

读者对象

使用 SDK 的软件开发工程师、项目经理和产品经理。

符号约定

在本文档中可能出现下列标识，代表的含义如下。

标识	说明
注意	表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
窍门	表示能帮助您解决某个问题或节省您的时间。
说明	表示是正文的附加信息，是对正文的强调和补充。

修订记录

版本号	修订内容	发布日期
V2.0.0	基础核心内容参考《NetSDK 编程指导手册》	2025.02
V1.0.0	首次发布。	2024.8

名词解释

以下对本文档中使用的专业名词分别说明，帮助您更好的理解各个业务功能。

名词	说明
主码流	视频码流类型的一种，一般是分辨率比较高，清晰度、画质更好的码流，在网络资源不受限的前提下能得到更好的体验。
辅码流	较主码流分辨率、清晰度都低一些，但占用的网络资源少；用户可以根据不同的适用场景选择不同的码流类型。
视频通道	设备的每路视频从 0 开始编号，得到的号码叫做视频通道号（简称通道），目前前端产品除热成像设备（TPC）外，一般只有一路通道，所以通道号一般为 0。
登录句柄	访问设备第一步需要登录（即鉴权），登录成功后获取到的唯一的 ID 号，标识此次登录会话的句柄，后续其他业务均需要用到该句柄，句柄直到用户登出（注销）失效。
相对定位	云台控制中快速定位的一种方法，通过提供云台坐标（X,Y 轴坐标）的差值给设备，设备根据当前云台位置和用户指定的差值，计算并转到最终位置；同时支持 ZOOM 控制。
绝对定位	云台控制中快速定位的一种方法，通过提供确定的水平、垂直坐标（角度坐标）给设备，设备直接转到用户指定的坐标位置，同时支持 ZOOM 控制。
PCM	脉冲编码调制（Pulse Code Modulation），是数字通信的编码方式之一，将模拟信号转换成数字信号的无损编码方式，适用于对数据传输率要求较高、需要更高宽带的用户使用。
PTZ	Pan Tilt Zoom，指云台全方位移动及镜头变倍、变焦控制。
热图	对应一个场景的温度分布数据，每个像素一个数据点。
云台	云台是承载摄像设备及防护罩并能够远程进行上下左右全方位控制的机械平台。云台实质上是由两个电机组成，可以实现水平和垂直的运动，从而给摄像机设备全方位、多角度的视野。

目录

前言 I
名词解释 I
第 1 章 内容简介 1
1.1 概述 1
1.2 适用性 2
第 2 章 主要功能 3
2.1 SDK 初始化 3
2.1.1 简介 3
2.1.2 接口总览 3
2.1.3 流程说明 3
2.1.4 示例代码 4
2.2 设备登录 5
2.2.1 简介 5
2.2.2 接口总览 5
2.2.3 流程说明 5
2.2.4 示例代码 7
2.3 实时预览 8
2.3.1 简介 8
2.3.2 接口总览 8
2.3.3 流程说明 8
2.3.4 示例代码 12
2.4 视频抓图 13
2.4.1 简介 13
2.4.2 接口总览 13
2.4.3 流程说明 13
2.4.4 示例代码 15
2.5 普通报警上报 16
2.5.1 简介 16
2.5.2 接口总览 16
2.5.3 流程说明 16
2.5.4 示例代码 17
2.6 智能事件订阅 19
2.6.1 简介 19
2.6.2 接口总览 19
2.6.3 流程说明 19
2.6.4 示例代码 20
2.7 热图（温度矩阵） 20
2.7.1 简介 20
2.7.2 接口总览 21
2.7.3 流程说明 21
2.7.4 示例代码 22
2.8 云台控制 23
2.8.1 预置点 23
2.8.2 云台绝对位置 25
2.8.3 云台持续控制 26
2.9 火情预警 29
2.9.1 简介 29

2.9.2 接口总览 29
2.9.3 流程说明 30
2.9.4 示例代码 32
2.10 冷热点获取 33
2.10.1 简介 33
2.10.2 接口总览 33
2.10.3 流程说明 33
2.10.4 示例代码 33
2.11 随机区域温度信息 34
2.11.1 简介 34
2.11.2 接口总览 34
2.11.3 流程说明 34
2.11.4 示例代码 35
2.12 测温设置 35
2.12.1 规则设置 35
2.12.2 全局设置 38
2.12.3 温度异常报警 40
2.12.4 获取规则温度信息 43
2.12.5 主动上报规则温度信息 44
第 3 章 接口说明 47
3.1 SDK 初始化 47
3.1.1 SDK 初始化 CLIENT_Init 47
3.1.2 SDK 清理 CLIENT_Cleanup 47
3.1.3 设置断线重连回调函数 CLIENT_SetAutoReconnect 47
3.1.4 设置网络参数 CLIENT_SetNetworkParam 47
3.2 设备登录 48
3.2.1 高安全级别登录 CLIENT_LoginWithHighLevelSecurity 48
3.2.2 用户登出设备 CLIENT_Logout 49
3.3 实时预览 49
3.3.1 打开预览 CLIENT_RealPlayEx 49
3.3.2 关闭预览 CLIENT_StopRealPlayEx 50
3.3.3 保存预览数据 CLIENT_SaveRealData 50
3.3.4 停止保存预览数据 CLIENT_StopSaveRealData 50
3.3.5 设置预览数据回调 CLIENT_SetRealDataCallBackEx2 50
3.4 视频抓图 51
3.4.1 设备抓图 CLIENT_SnapPictureToFile 51
3.4.2 抓图 CLIENT_CapturePictureEx 51
3.5 普通报警上报 52
3.5.1 设置报警回调函数 CLIENT_SetDVRMessCallBack 52
3.5.2 订阅报警 CLIENT_StartListenEx 52
3.5.3 停止订阅报警 CLIENT_StopListen 53
3.6 智能事件订阅 53
3.6.1 开始智能事件订阅 CLIENT_RealLoadPictureEx 53
3.6.2 停止智能事件订阅 CLIENT_StopLoadPic 54
3.7 热图 54
3.7.1 订阅温度分布数据 CLIENT_RadiometryAttach 54
3.7.2 取消订阅温度分布数据 CLIENT_RadiometryDetach 55
3.7.3 获取温度分布数据 CLIENT_RadiometryFetch 55
3.7.4 解析温度分布数据 CLIENT_RadiometryDataParse 55
3.8 云台控制 56
3.8.1 查询设备状态 CLIENT_QueryDevState 56

3.8.2 云台控制 CLIENT_DHPTZControlEx2 56
3.9 火情预警 57
3.9.1 订阅火点报警事件 CLIENT_RealLoadPictureEx 57
3.9.2 停止订阅火点报警事件 CLIENT_StopLoadPic 57
3.9.3 设置火点信息回调函数接口 CLIENT_SetDVRMessCallBack 58
3.9.4 订阅火点信息接口 CLIENT_StartListenEx 58
3.9.5 停止订阅火点信息接口 CLIENT_StopListen 58
3.10 冷热点获取 RadiometryGetCurrentHotColdSpotInfo 58
3.11 随机区域温度信息 RadiometryGetRandomRegionTemper 59
3.12 测温设置 59
3.12.1 设备配置信息获取 CLIENT_GetNewDevConfig 59
3.12.2 解析查询到的配置信息 CLIENT_ParseData 60
3.12.3 组成要设置的配置信息 CLIENT_PacketData 60
3.12.4 设备配置信息设置 CLIENT_SetNewDevConfig 61
3.12.5 设置报警回调函数 CLIENT_SetDVRMessCallBack 61
3.12.6 订阅报警 CLIENT_StartListenEx 62
3.12.7 停止订阅报警 CLIENT_StopListen 63
3.12.8 获取规则温度信息 CLIENT_RadiometryGetCurTemperAll 63
3.12.9 订阅测温温度数据 CLIENT_RadiometryAttachTemper 63
3.12.10 取消订阅测温温度数据 CLIENT_RadiometryAttachTemper 64
第 4 章 回调函数 65
4.1 断线回调函数 fDisConnect 65
4.2 断线重连回调函数 fHaveReConnect 65
4.3 实时预览数据回调函数 fRealDataCallBackEx 65
4.4 普通报警回调函数 fMessCallBack 66
4.5 智能事件回调 fAnalyzerDataCallBack 67
4.6 温度分布数据回调函数 fRadiometryAttachCB 67
附录 1 法律声明 68
附录 2 安全建议 69

第 1 章 内容简介

1.1 概述

本文档主要介绍 SDK 接口参考信息，包括主要功能、接口函数和回调函数。

主要功能包括：SDK 初始化、设备登录、实时预览、云台控制、视频抓图和热图。

根据环境不同，开发包包含的文件会不同，具体如下。

- Windows 开发包所包含的文件，请参见表 1-1。

表1-1 Windows 开发包包括的文件

库类型	库文件名称	库文件说明
功能库	dhnetsdk.h	头文件
	dhnetsdk.lib	Lib 文件
	dhnetsdk.dll	库文件
	avnetsdk.dll	库文件
配置库	avglobal.h	头文件
	dhconfigsdk.h	配置头文件
	dhconfigsdk.lib	Lib 文件
	dhconfigsdk.dll	库文件
播放（编码解码）辅助库	dhplay.dll	大华播放库
dhnetsdk 辅助库	lvsDrawer.dll	图像显示库
	StreamConvertor.dll	转码库

- Linux 开发包所包含的文件，请参见表 1-2。

表1-2 Linux 开发包包括的文件

库类型	库文件名称	库文件说明
功能库	dhnetsdk.h	头文件
	libdhnetsdk.so	库文件
	libavnetsdk.so	库文件
配置库	avglobal.h	头文件
	dhconfigsdk.h	配置头文件
	libdhconfigsdk.so	配置库
libdhnetsdk.so 辅助库	libStreamConvertor.so	转码库

说明

- SDK 的功能库和配置库是必备库。
- 功能库是设备网络 SDK 的主体，主要用于网络客户端与各类产品之间的通讯交互，负责远程控制、查询、配置及码流数据的获取和处理等。
- 配置库针对配置功能的结构体进行打包和解析。
- 推荐使用播放库进行码流解析和播放。
- 辅助库用于预览、回放、对讲等功能的音视频码流解码以及本地音频采集。

1.2 适用性

- 推荐内存：不低于 512M。
- SDK 支持的系统：
  - ◇ Windows
    Windows 10/Windows 8.1/Windows 7 以及 Windows Server 2008/2003。
  - ◇ Linux
    Red Hat/SUSE 等通用 Linux 系统。

第 2 章 主要功能

2.1 SDK 初始化

2.1.1 简介

初始化是 SDK 进行各种业务的第一步。初始化本身不包含监控业务，但会设置一些影响全局业务的参数。

- SDK 的初始化将会占用一定的内存。
- 同一个进程内，只有第一次初始化有效。
- 使用完毕后需要调用 SDK 清理接口以释放资源。

2.1.2 接口总览

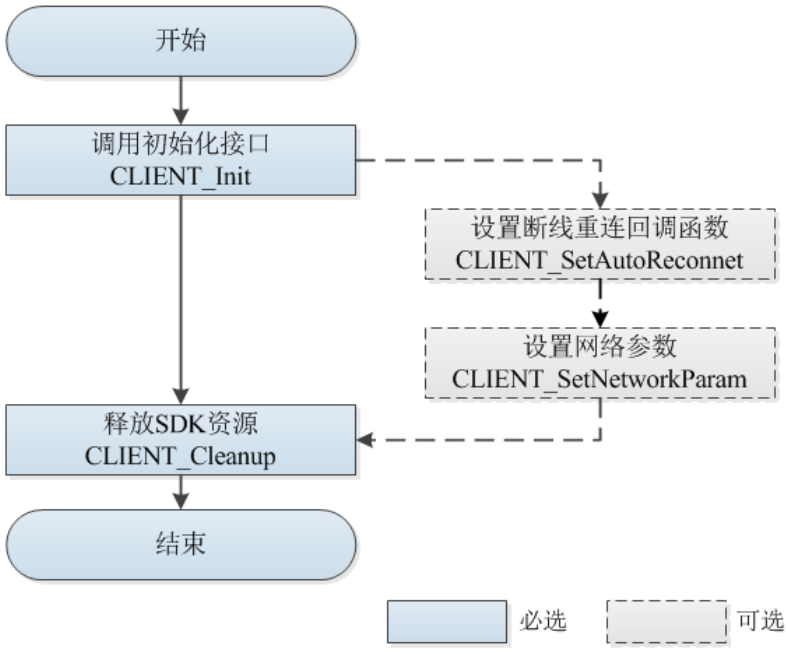
表2-1 SDK 初始化接口信息

接口	说明
CLIENT_Init	SDK 初始化接口
CLIENT_Cleanup	SDK 清理接口
CLIENT_SetAutoReconnect	设置断线重连回调接口
CLIENT_SetNetworkParam	设置登录网络环境接口

2.1.3 流程说明

SDK 初始化业务流程如图 2-1 所示。

图2-1 SDK 初始化业务流程



流程说明

步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。

步骤2　（可选）调用 CLIENT_SetAutoReconnect 设置断线重连回调函数，设置后 SDK 内部断线自动重连。

步骤3　（可选）调用 CLIENT_SetNetworkParam 设置网络登录参数，参数中包含登录设备超时时间和尝试次数。

步骤4　SDK 所有功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- SDK 的 CLIENT_Init 和 CLIENT_Cleanup 接口需成对调用，支持单线程多次成对调用，但建议全局调用一次。
- 初始化：CLIENT_Init 接口内部多次调用时，仅在内部用做计数，不会重复申请资源。
- 清理：CLIENT_Cleanup 接口内会清理所有已开启的业务，如登录、实时预览和报警订阅等。
- 断线重连：SDK 可以设置断线重连功能，当遇到一些特殊情况（例如断网、断电等）设备断线时，在 SDK 内部会定时持续不断地进行登录操作，直至成功登录设备。断线重连后可以恢复实时预览和录像回放业务，其他业务无法恢复。

2.1.4 示例代码

```
// 通过 CLIENT_Init 设置该回调函数，当设备出现断线时，SDK 通过该函数通知用户
void CALLBACK DisConnectFunc(LONG lLoginID, char *pchDVRIP, LONG nDVRPort, DWORD dwUser)
{
    printf("Call DisConnectFunc: lLoginID[0x%x]\n", lLoginID);
}
// 初始化 SDK
BOOL bNetSDKInitFlag = CLIENT_Init(DisConnectFunc, 0);
if (FALSE == bNetSDKInitFlag)
{
    printf("Initialize client SDK fail; \n");
    return -1;
}
// 清理初始化资源
if (TRUE == bNetSDKInitFlag)
{
    CLIENT_Cleanup();
}
```

2.2 设备登录

2.2.1 简介

设备登录，即用户鉴权，是进行其他业务的前提。

用户登录设备产生唯一的登录 ID，其他功能的 SDK 接口需要传入登录 ID 才可执行。登出设备后，登录 ID 失效。

2.2.2 接口总览

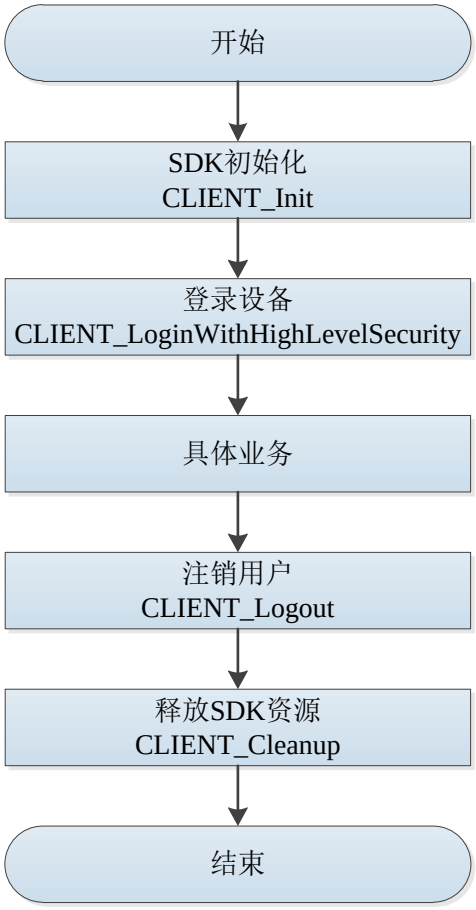
表2-2 设备登录接口信息

接口	说明
CLIENT_LoginWithHighLevelSecurity	高安全级别登录接口。 **说明** CLIENT_LoginEx2 仍然可以使用，但存在安全风险。所以强烈推荐使用最新接口 CLIENT_LoginWithHighLevelSecurity 登录设备。
CLIENT_Logout	登出接口

2.2.3 流程说明

登录业务流程如图 2-2 所示。

图2-2 登录业务流程



流程说明

步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。
步骤2　调用 CLIENT_LoginWithHighLevelSecurity 登录设备。
步骤3　登录成功后，用户可以实现需要的业务功能。
步骤4　业务使用完后，调用 CLIENT_Logout 登出设备。
步骤5　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 登录句柄：登录成功时接口返回值非 0（即句柄可能小于 0，也属于登录成功）；同一设备登录多次，每次的登录句柄不一样。如果无特殊业务，建议只登录一次，登录的句柄可以重复用于其他各种业务。
- 登出：接口内部会释放登录会话中已打开的业务，但建议用户不要依赖登出接口的清理功能。例如打开预览后，在不需要使用预览时，用户应该调用结束预览的接口。
- 登录与登出配对使用，登录会消耗一定的内存和 socket 信息，在登出后释放资源。
- 登录失败：建议通过登录接口的 error 参数（登录错误码）初步排查。常见错误码请参见表 2-3。

表2-3 常见错误码

error 的错误码	对应的含义
1	密码不正确
2	用户名不存在
3	登录超时

error 的错误码	对应的含义
4	账号已登录
5	账号已被锁定
6	账号被列为黑名单
7	资源不足，设备系统忙
8	子连接失败
9	主连接失败
10	超过最大用户连接数
11	缺少 avnetsdk 或 avnetsdk 的依赖库
12	设备未插入 U 盘或 U 盘信息错误
13	客户端 IP 地址没有登录权限

更多错误码信息请参见《网络 SDK 开发手册》中的“CLIENT_LoginWithHighLevelSecurity 接口”描述。其中错误码 3 规避示例代码如下：

```
NET_PARAM stuNetParam = {0};
stuNetParam.nWaittime = 8000; // unit ms
CLIENT_SetNetworkParam (&stuNetParam);
```

2.2.4 示例代码

```
NET_DEVICEINFO_Ex stDevInfo = {0};
int nError = 0;
// 登录设备
NET_IN_LOGIN_WITH_HIGHLEVEL_SECURITY stInparam;
memset(&stInparam, 0, sizeof(stInparam));
stInparam.dwSize = sizeof(stInparam);
strncpy(stInparam.szIP, csIp.GetBuffer(0), sizeof(stInparam.szIP) - 1);
strncpy(stInparam.szPassword, csPwd.GetBuffer(0), sizeof(stInparam.szPassword) - 1);
strncpy(stInparam.szUserName, csName.GetBuffer(0), sizeof(stInparam.szUserName) - 1);
stInparam.nPort = sPort;
stInparam.emSpecCap = EM_LOGIN_SPEC_CAP_TCP;
NET_OUT_LOGIN_WITH_HIGHLEVEL_SECURITY stOutparam;
memset(&stOutparam, 0, sizeof(stOutparam));
stOutparam.dwSize = sizeof(stOutparam);


LLONG lLoginHandle = CLIENT_LoginWithHighLevelSecurity(&stInparam, &stOutparam);
// 登出设备
if (0 != lLoginHandle)
{
    CLIENT_Logout(lLoginHandle);
}
```

2.3 实时预览

2.3.1 简介

实时预览，即向存储设备或前端设备获取实时码流的功能，是监控系统的重要组成部分。

SDK 登录设备后，可向设备获取主码流和辅码流。

- 支持用户传入窗口句柄，SDK 直接进行码流解析及播放（此功能仅限 Windows 版本）。
- 支持回调实时码流数据给用户，让用户自己处理。
- 支持保存实时录像到指定文件，用户可通过自行保存回调码流实现，也可以通过调用 SDK 接口实现。

2.3.2 接口总览

表2-4 实时预览接口信息

接口	说明
CLIENT_RealPlayEx	开始实时预览扩展接口
CLIENT_StopRealPlayEx	停止实时预览扩展接口
CLIENT_SaveRealData	开始本地保存实时预览数据
CLIENT_StopSaveRealData	停止本地保存实时预览数据
CLIENT_SetRealDataCallBackEx2	设置实时预览数据回调函数扩展接口

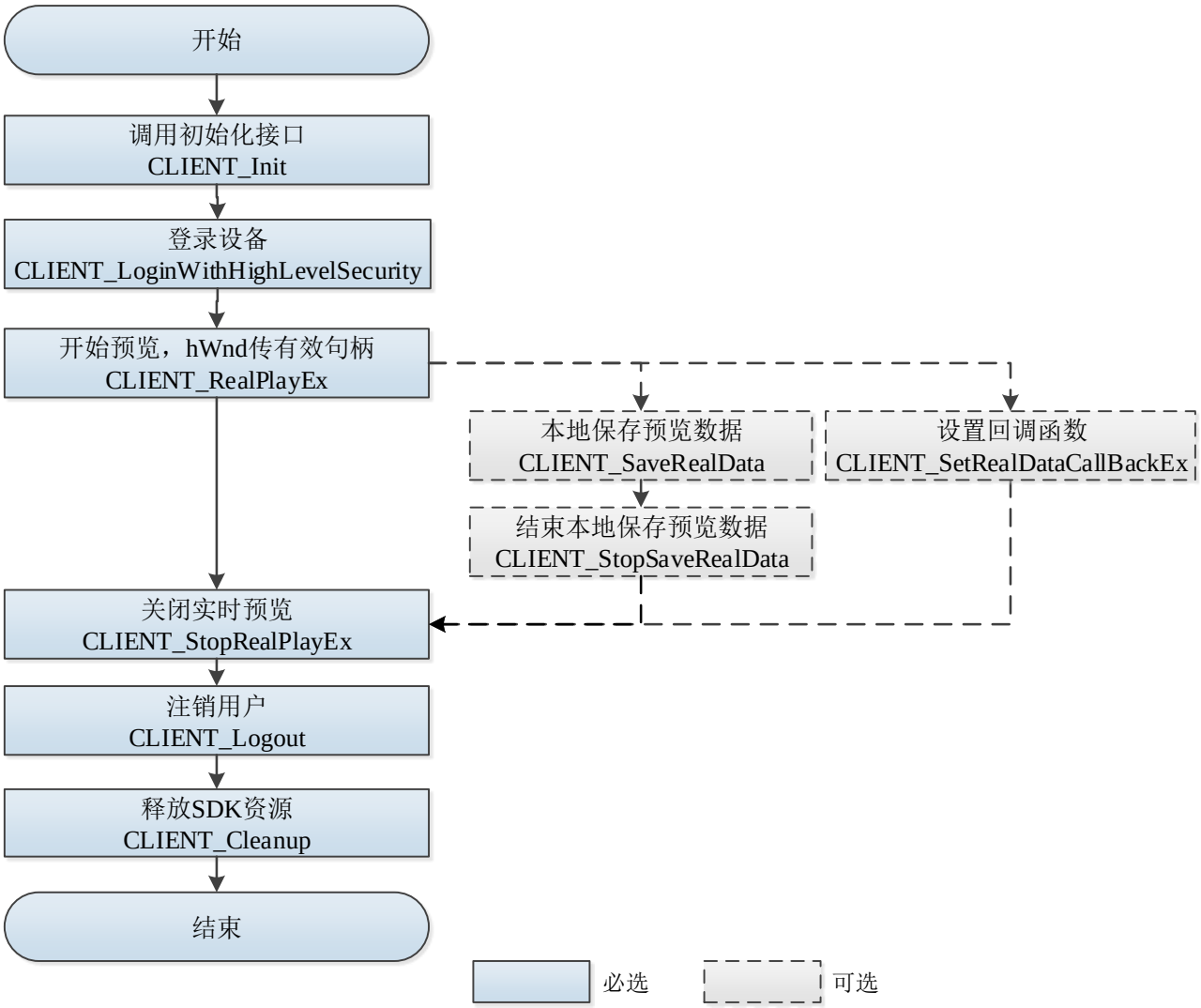
2.3.3 流程说明

实时监控的实现方式有两种，分别为 SDK 集成播放库进行播放及用户自己调用播放库播放码流方式进行播放。

2.3.3.1 SDK 解码播放

SDK 内部调用辅助库里的 PlaySDK 库实现实时播放。SDK 解码播放流程如图 2-3 所示。

图2-3 SDK 解码播放流程图



流程说明

步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。
步骤2　调用 CLIENT_LoginWithHighLevelSecurity 登录设备。
步骤3　调用 CLIENT_RealPlayEx 启动实时预览，参数 **hWnd** 为有效窗口句柄。
步骤4　（可选）调用 CLIENT_SaveRealData 开始保存预览数据。
步骤5　（可选）调用 CLIENT_StopSaveRealData 结束保存，生成本地视频文件。
步骤6　（可选）若调用 CLIENT_SetRealDataCallBackEx2，用户可将视频数据选择保存或转发。若保存成文件，与步骤 4、5 效果相同。
步骤7　实时预览使用完毕后，调用 CLIENT_StopRealPlayEx 停止实时预览。
步骤8　业务使用完后，调用 CLIENT_Logout 登出设备。
步骤9　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- SDK 解码播放只支持 Windows 系统，非 Windows 系统需要用户获取码流后自己调用解码显示。

- 多线程调用：同一个登录会话内的业务，不支持多线程调用；但可以多个线程处理不同的登录会话中的业务，但不建议这样调用。
- 超时：接口内申请预览资源需和设备做一些约定，然后才请求预览数据，过程中有一些超时时间的设定（请参见 NET_PARAM 结构体），其中与预览相关的字段为 **nGetConnInfoTime**。如果实际使用中（如网络状况不良）有超时现象，可将 **nGetConnInfoTime** 的值修改大一些。示例代码如下，在 CLIENT_Init 函数后调用，调用一次即可：

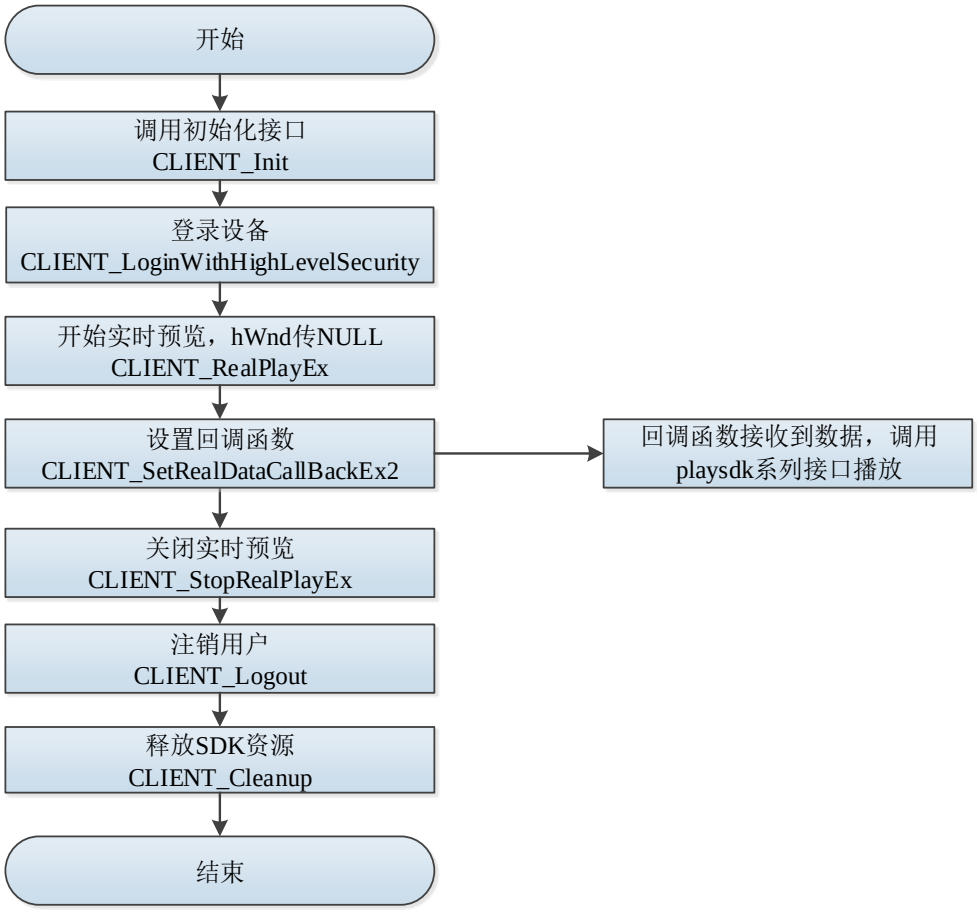
```
NET_PARAM stuNetParam = {0};
stuNetParam. nGetConnInfoTime = 5000; //单位 ms
CLIENT_SetNetworkParam (&stuNetParam);
```

- 重复打开失败：部分设备不支持同一次登录下同一个通道多次打开，当重复打开同一通道的预览，可能会出现第一次打开成功，后续打开失败的现象。建议：
  - ◇ 将已打开的通道先关闭。例如已经开启通道一的主码流视频，希望再打开通道一的辅码流视频时，可先关闭通道一的主码流视频，再开启通道一的辅码流视频。
  - ◇ 登录两次设备获取两个登录句柄，分别处理主码流和辅码流业务。
- 接口成功无画面：SDK 内部解码需要使用到 dhplay.dll，建议查看运行目录下是否缺少 dhplay.dll 及其依赖的辅助库，具体请参见表 1-1。
- 系统资源不足的情况下，设备可能返回错误而不恢复码流，可以在报警回调函数（即 CLIENT_SetDVRMessCallBack 中设置的回调函数）收到事件 DH_REALPLAY_FAILD_EVENT，该事件包含了详细的错误码，请参见《网络 SDK 开发手册》中的“DEV_PLAY_RESULT 结构体”。
- 32 路限制：解码显示比较消耗资源，特别是高分辨率视频，考虑到客户端硬件资源有限，一般同时解码显示的通道数有限，所以该方式暂时限定为最多 32 路，如超过 32 路，建议使用“2.3.3.2 调用第三方解码播放库”。

2.3.3.2 调用第三方解码播放库

SDK 回调实时预览码流给用户，用户调用 PlaySDK 进行解码播放。用户第三方解码播放流程如图 2-4 所示。

图2-4 第三方解码播放流程图



流程说明

步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。
步骤2　调用 CLIENT_LoginWithHighLevelSecurity 登录设备。
步骤3　登录成功后，调用 CLIENT_RealPlayEx 启动实时预览，参数 **hWnd** 为 **NULL**。
步骤4　调用 CLIENT_SetRealDataCallBackEx2 设置实时数据回调函数。
步骤5　在回调函数中将数据传给 PlaySDK 完成解码。
步骤6　实时预览使用完毕后，调用 CLIENT_StopRealPlayEx 停止实时预览。
步骤7　业务使用完后，调用 CLIENT_Logout 登出设备。
步骤8　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 码流格式：推荐使用 PlaySDK 解码。
- 画面卡顿：
  - ◇ 使用 PlaySDK 解码时，解码通道缓存大小有默认值（PlaySDK 中的 PLAY_OpenStream 接口）。如果码流的分辨率很大，建议修改参数值，例如改为 3M。
  - ◇ SDK 回调函数需用户返回后才能回调下一段，建议用户在回调中不要做耗时操作，否则会严重影响性能。

2.3.4 示例代码

2.3.4.1 SDK 解码播放

```
//以开启第一路的主码流预览为例，hWnd 为界面窗口句柄
LLONG lRealHandle = CLIENT_RealPlayEx(lLoginHandle, 0, hWnd, DH_RType_Realplay);
if (NULL == lRealHandle)
{
    printf("CLIENT_RealPlayEx: failed! Error code: %x.\n", CLIENT_GetLastError());
}
printf("input any key to quit!\n");
getchar();
// 关闭预览
if (NULL != lRealHandle)
{
CLIENT_StopRealPlayEx(lRealHandle);
}
```

2.3.4.2 调用播放库

```
void CALLBACK RealDataCallBackEx(LLONG lRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, LLONG param, LDWORD dwUser);
//以开启第一路的主码流预览为例
LLONG lRealHandle = CLIENT_RealPlayEx(lLoginHandle, 0, NULL, DH_RType_Realplay);
if (NULL == lRealHandle)
{
printf("CLIENT_RealPlayEx: failed! Error code: %x.\n", CLIENT_GetLastError());
}
else
{
DWORD dwFlag = REALDATA_FLAG_RAW_DATA; //原始数据标志
    CLIENT_SetRealDataCallBackEx2(lRealHandle, &RealDataCallBackEx, NULL, dwFlag);
}

printf("input any key to quit!\n");
getchar();
// 关闭预览
if (0 != lRealHandle)
{
    CLIENT_StopRealPlayEx(lRealHandle);
```

```
}

void CALLBACK RealDataCallBackEx(LLONG lRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, LLONG param, LDWORD dwUser)
{
    // 从设备获取的码流数据，需调用 PlaySDK 的接口，详见 SDK 预览 demo 源码
    printf("receive real data, param: lRealHandle[%p], dwDataType[%d], pBuffer[%p], dwBufSize[%d]\n", lRealHandle, dwDataType, pBuffer, dwBufSize);
}
```

2.4 视频抓图

2.4.1 简介

视频抓图，即获取前端产品当前画面的图片数据。本章介绍以下两种抓图方式：

- 网络抓图：用户调用 SDK 接口，接口内部发送抓图命令给设备，设备抓取当前画面并通过网络发送给 SDK，SDK 将接收到的图片数据返回给用户。
- 本地抓图：用户在已打开预览的前提下，可以将监控中的数据保存为图片，该图片是从显示画面中保存帧信息，与设备之间没有网络交互。

2.4.2 接口总览

表2-5 视频抓图的接口信息

接口	说明
CLIENT_SnapPictureToFile	抓图同步接口，将图片数据直接返回给用户。
CLIENT_CapturePictureEx	本地抓图，参数可以是预览或回放的句柄。

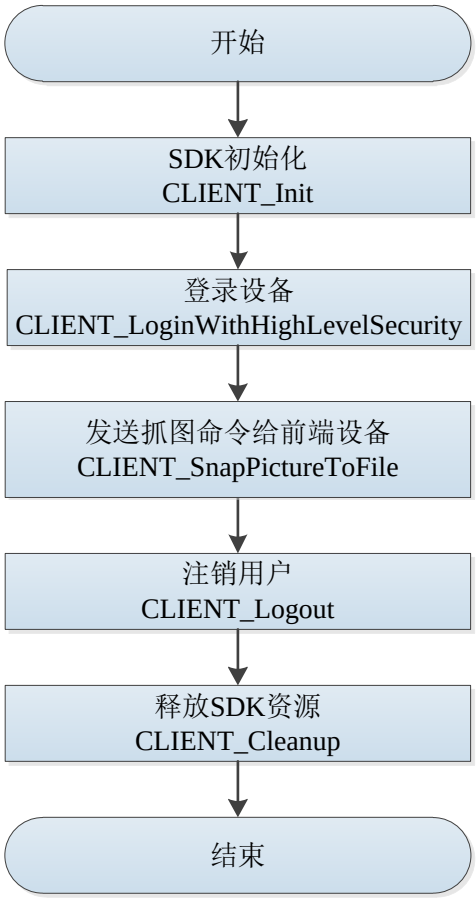
2.4.3 流程说明

视频抓图可分为网络抓图和本地抓图两种。

2.4.3.1 网络抓图

网络抓图流程如图 2-5 所示。

图2-5 网络抓图流程图



流程说明

步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。
步骤2　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。
步骤3　调用 CLIENT_SnapPictureToFile 获取图片信息。
步骤4　调用 CLIENT_Logout 登出设备。
步骤5　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 图片大小限制：SDK 内部分配了固定大小的内存来接收设备端返回的图片数据，如果图片超过默认分配的大小，SDK 返回截断后的图片数据。
  SDK 提供了修改默认内存大小的接口，如果实际使用中有图片（如图像分辨率很高）被截断的现象，可以将 **nPicBufSize** 的值修改大一些。示例代码如下，在 CLIENT_Init 函数后调用，调用一次即可：

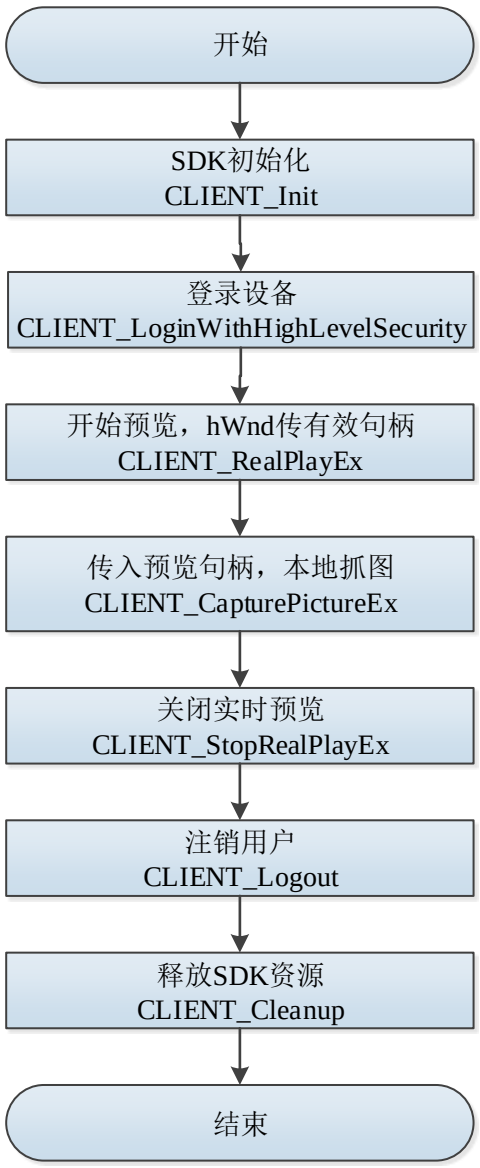
```
NET_PARAM stuNetParam = {0};
stuNetParam. nPicBufSize = 4*1024*1024; // unit byte
CLIENT_SetNetworkParam (&stuNetParam);
```

- 多线程调用：同一个登录会话内的业务，不支持多线程调用。
- 抓图配置：网络抓图的图片质量、分辨率等可以设置的。但是如果默认配置效果能满足，建议不用修改，示例源码详见官网发布包。
- 图片保存形式：图片数据以内存的形式返回，同时接口支持保存成文件（前提是用户已设置 NET_IN_SNAP_PIC_TO_FILE_PARAM 的 **szFilePath** 字段）。

2.4.3.2 本地抓图

本地抓图流程如图 2-6 所示。

图2-6 本地抓图流程图



流程说明

步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。
步骤2　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。
步骤3　调用 CLIENT_RealPlayEx 开启预览，获取预览句柄。
步骤4　调用 CLIENT_CapturePictureEx 传入预览句柄。
步骤5　调用 CLIENT_StopRealPlayEx 关闭实时预览。
步骤6　调用 CLIENT_Logout 登出设备。
步骤7　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

2.4.4 示例代码

```
// 网络抓图示例
```

```
NET_IN_SNAP_PIC_TO_FILE_PARAM stuInParam = {sizeof(stuInParam)};
NET_OUT_SNAP_PIC_TO_FILE_PARAM stuOutParam = {sizeof(stuOutParam)};
SNAP_PARAMS stuSnapParams = {0};
stuSnapParams.Channel = 0;     // 以第一个通道为例
int nBufferLen = 2*1024*1024;
char* pBuffer = new char[nBufferLen]; // 图片缓存
memset(pBuffer, 0, nBufferLen);
stuOutParam.szPicBuf = pBuffer;
stuOutParam.dwPicBufLen = nBufferLen;
if (FALSE == CLIENT_SnapPictureToFile(lLoginHandle, &stuSnapParams))
{
printf("CLIENT_SnapPictureToFile Failed!Last Error[%x]\n", CLIENT_GetLastError());
}
delete[] pBuffer;

// 本地抓图示例，hPlayHandle 为打开预览获取的句柄
if (FALSE == CLIENT_CapturePictureEx(hPlayHandle, "test.jpg", NET_CAPTURE_JPEG))
{
printf("CLIENT_CapturePictureEx Failed!Last Error[%x]\n", CLIENT_GetLastError());
}
```

2.5 普通报警上报

2.5.1 简介

报警上报实现方式为：通过 SDK 登录设备并向设备订阅报警功能，设备检测到报警事件立即发送给 SDK，通过报警回调函数即可获取相应报警信息。

用户在订阅行为检测事件前，需要在设备 Web 端配置行为检测触发规则。

2.5.2 接口总览

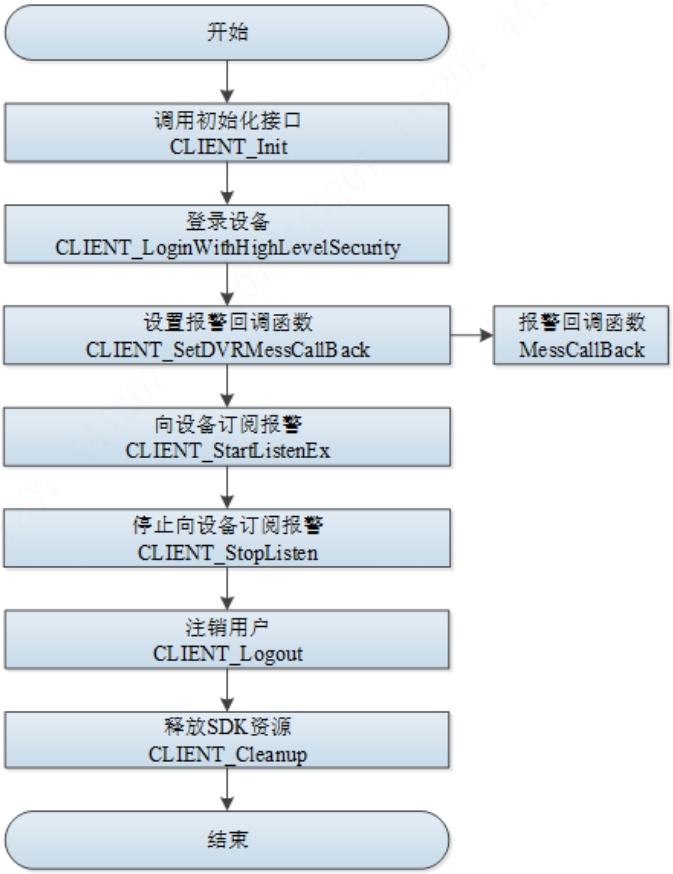
表2-6 报警上报的接口信息

接口	说明
CLIENT_SetDVRMessCallBack	设置报警回调函数接口
CLIENT_StartListenEx	订阅报警扩展接口
CLIENT_StopListen	停止订阅报警

2.5.3 流程说明

报警上报流程如图 2-7 所示。

图2-7 报警上报流程



流程说明

步骤1　调用 CLIENT_Init 函数，完成 SDK 初始化流程。

步骤2　调用 CLIENT_LoginWithHighLevelSecurity 函数登录设备。

步骤3　调用 CLIENT_SetDVRMessCallBack 函数，设置报警回调函数，该接口需在报警订阅之前调用。

步骤4　调用 CLIENT_StartListenEx 函数，向设备订阅报警。订阅成功后，设备上报的报警事件通过 CLIENT_SetDVRMessCallBack 函数设置的回调函数通知用户。

步骤5　报警上报功能使用完毕后，调用 CLIENT_StopListen 函数停止向设备订阅报警。

步骤6　调用 CLIENT_Logout 函数登出设备。

步骤7　SDK 功能使用完后，调用 CLIENT_Cleanup 函数释放 SDK 资源。

注意事项

如果之前能上报的报警突然不再上报了，则需排查下设备是否断线。

2.5.4 示例代码

```
// 报警回调
int CALLBACK  afMessCallBack(LONG lCommand, LLONG lLinID, char *pBuf, DWORD dwBufLen,
char *pchDVRIP, LONG nDVRPort, LDWORD dwUser)
```

```
{
if(lCommand == DH_ALARM_FIREWARNING_INFO)// 热成像火情信息上报
{
        printf("热成像火情信息上报");
ALARM_FIREWARNING_INFO_DETAIL*  pstStorageInfo  =
(ALARM_FIREWARNING_INFO_DETAIL*)pBuf;
    //接下来可以通过 pstStorageInfo 取对应的报警信息。
}
}

// 设置报警回调
CLIENT_SetDVRMessCallBack(&afMessCallBack,0);
// 订阅报警
BOOL bRet = CLIENT_StartListenEx(lLoginHandle);
if(!bRet)
{
    printf("CLIENT_StartListenEx Failed, Last Error[%x]\n" ,
    CLIENT_GetLastError());
}
else
{
printf("listen succeed.\n");
}

// 停止报警订阅
BOOL bRet = CLIENT_StopListen(lLoginHandle);
if(!bRet)
{
    printf("CLIENT_StopListen Failed, Last Error[%x]\n" ,
    CLIENT_GetLastError());
}
else
{
printf("stop listen succeed.\n");
}
```

2.6 智能事件订阅

2.6.1 简介

热成像智能事件上报实现方式为 SDK 主动连接设备，并向设备订阅智能事件功能，设备检测到智能事件立即发送给 SDK。

用户在订阅行为检测事件前，需要在设备 Web 端配置行为检测触发规则。

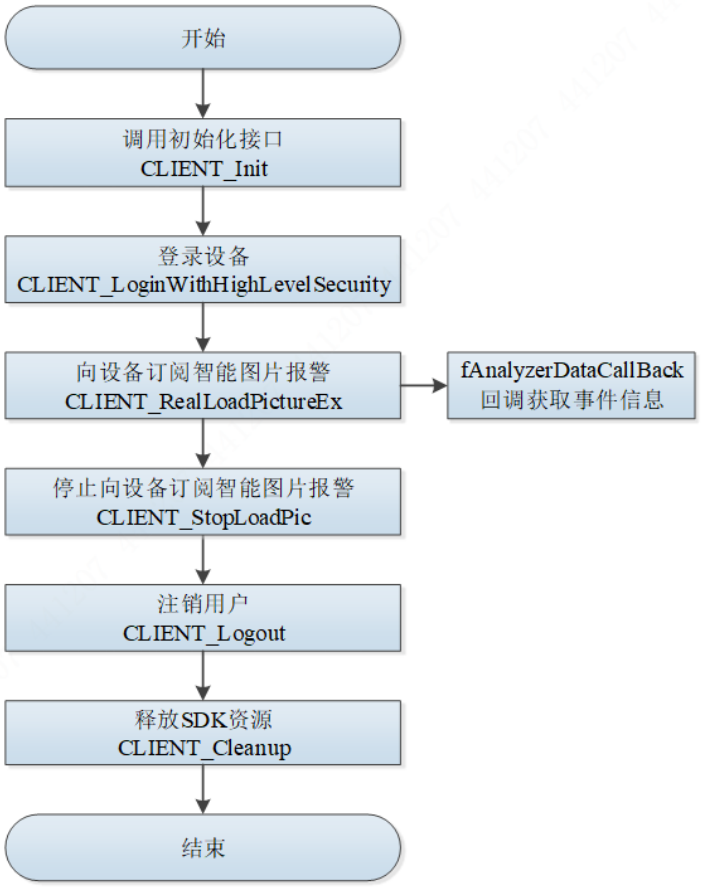
2.6.2 接口总览

接口	说明
CLIENT_RealLoadPictureEx	订阅智能事件
CLIENT_StopLoadPic	取消订阅智能事件

2.6.3 流程说明

事件上报流程，如图 2-8 所示。

图2-8 事件上报流程



流程说明

步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。

步骤2　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。

步骤3　调用 CLIENT_RealLoadPictureEx 向设备订阅智能事件。

步骤4　订阅成功后，设备上报的智能事件通过 fAnalyzerDataCallBack 回调函数上报智能事件。通过该函数，可以根据报警类型过滤出您需要的智能报警事件。

步骤5　智能事件上报功能使用完毕后，调用 CLIENT_StopLoadPic 停止订阅智能事件。
步骤6　业务使用完后，调用 CLIENT_Logout 登出设备。
步骤7　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 订阅事件类型：如果需要同时上报不同智能事件时，支持订阅所有智能事件（EVENT_IVS_ALL）；也支持订阅单个智能事件。
- 设置接收图片缓存：由于 SDK 默认接收缓存是 2M，当回调上来的图片数据大于 2M 时，用户需调用 CLIENT_SetNetworkParam 接口设置接收图片缓存，否则将丢弃大于 2M 的数据包。
- 设置是否接收图片：由于一些设备所在网络环境是 3G 或 4G 网络，当 SDK 连接设备时，如不需要接收图片可以把 CLIENT_RealLoadPictureEx 接口中 bNeedPicFile 参数设置为 False，只接收人脸事件信息，不带图片。

2.6.4 示例代码

```
// 智能事件上报回调函数
int CALLBACK AnalyzerDataCallBack(LLONG lAnalyzerHandle, DWORD dwAlarmType, void* pAlarmInfo, BYTE *pBuffer, DWORD dwBufSize, LDWORD dwUser, int nSequence, void *reserved)
{
switch(dwAlarmType)
{
// 过滤出你想要的智能事件
.....
case EVENT_IVS_FIREDETECTION: //火警事件
.....
default:
break;
}
}

// 订阅智能事件上报
LLONG lAnalyerHandle = CLIENT_RealLoadPictureEx(lLoginHandle, 0, (DWORD)EVENT_IVS_ALL, TRUE, AnalyzerDataCallBack, NULL, NULL);
if(NULL == lAnalyerHandle)
{
printf("CLIENT_RealLoadPictureEx: failed! Error code %x.\n", CLIENT_GetLastError());
return -1;
}

// 取消订阅智能事件上报
CLIENT_StopLoadPic(lAnalyerHandle);
```

2.7 热图（温度矩阵）

2.7.1 简介

热图功能可以获取画面的温度分布数据，并进一步获取灰度图和温度图，即图片每个像素点的灰度和温度。

仅带测温功能的设备支持此功能。

2.7.2 接口总览

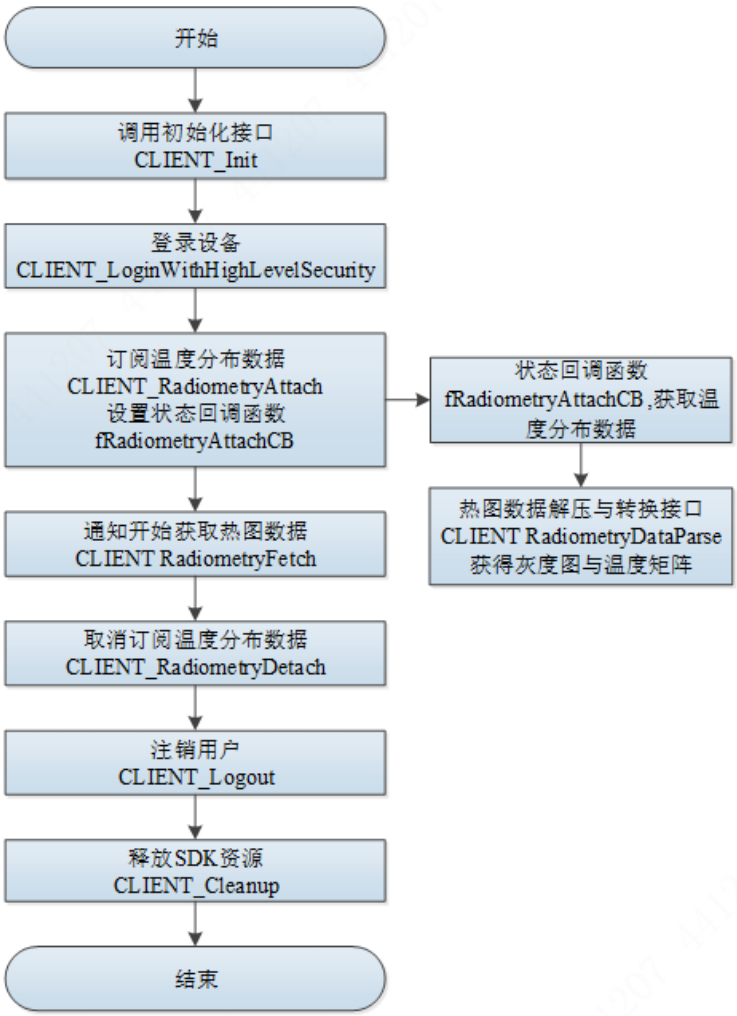
表2-7 热图接口信息

接口	说明
CLIENT_RadiometryAttach	订阅温度分布数据（热图）。
CLIENT_RadiometryDetach	取消订阅温度分布数据。
CLIENT_RadiometryFetch	通知开始获取热图数据。
CLIENT_RadiometryDataParse	热图数据解压与转换接口，热图数据转换为以像素点为单位的灰度和温度数据。

2.7.3 流程说明

流程如图 2-9 所示。

图2-9 热图流程图



流程说明

步骤1　完成 SDK 初始化流程。

步骤2　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。

步骤3　调用 CLIENT_RadiometryAttach 订阅温度分布数据，注册状态回调函数，设备上报温度状态后，回调函数 fRadiometryAttachCB 会提示用户

步骤4　调用 CLIENT_RadiometryFetch 通知设备开始获取热图数据，每次需要热图数据时调用一次；

步骤5　如果收到温度状态信息，可以调用 CLIENT_RadiometryDataParse 获得每个像素的灰度和温度值

步骤6　使用完毕后调用 CLIENT_RadiometryDetach 取消订阅温度分布数据

步骤7　调用 CLIENT_Logout，登出设备

步骤8　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 无热图数据上报：调用 CLIENT_RadiometryFetch 才会开始获取热图数据。如果在收到热图数据之前调用 CLIENT_RadiometryDetach，也不会收到热图数据。
- 订阅通道号：如果摄像机是单目摄像机，则订阅通道号为 0，如果摄像机为双目摄像机，则订阅通道号为 0 或者 1。双目热成像摄像机第二个通道是热成像。

2.7.4 示例代码

```
// 温度状态回调函数
void CALLBACK cbRadiometryAttachCB(LLONG lAttachHandle, NET_RADIOMETRY_DATA* pBuf, int nBufLen, LDWORD dwUser)
{
    int nPixel = pBuf->stMetaData.nWidth*pBuf->stMetaData.nHeight;
    unsigned short *pGray = new unsigned short[nPixel];
    memset(pGray,0,nPixel);

    float *pTemp = new float[nPixel];
    memset(pTemp,0,nPixel);

    CLIENT_RadiometryDataParse(pBuf,pGray,pTemp);
    delete[] pGray;
    delete[] pTemp;
}

// 订阅热图
NET_IN_RADIOMETRY_ATTACH stIn = {sizeof(stIn), 1, cbRadiometryAttachCB};
NET_OUT_RADIOMETRY_ATTACH stOut = {sizeof(stOut)};

LLONG attachHandle = CLIENT_RadiometryAttach(loginId, &stIn, &stOut, 3000);
if (NULL == attachHandle)
{
    // 订阅失败
}

// 通知设备开始采集数据
NET_IN_RADIOMETRY_FETCH stInFetch = {sizeof(stInFetch), 1};
NET_OUT_RADIOMETRY_FETCH stOutFetch = {sizeof(stOutFetch)};
CLIENT_RadiometryFetch(m_lLoginID, &stInFetch, &stOutFetch, 3000);

// 使用完毕，取消订阅热图
CLIENT_RadiometryDetach(attachHandle);
```

2.8 云台控制

云台是指承载摄像设备及防护罩并能够远程进行全方位控制的机械平台。云台实质上是由两个电机组成，可以实现水平和垂直的运动，从而给摄像机设备全方位、多角度的视野。

本节主要指导用户如何通过 SDK 实现方向控制（简称八方位控制，具体包括上、下、左、右，左上、左下、右上和右下）、聚焦、变倍、光圈、快速定位和精确三维定位功能。

2.8.1 预置点

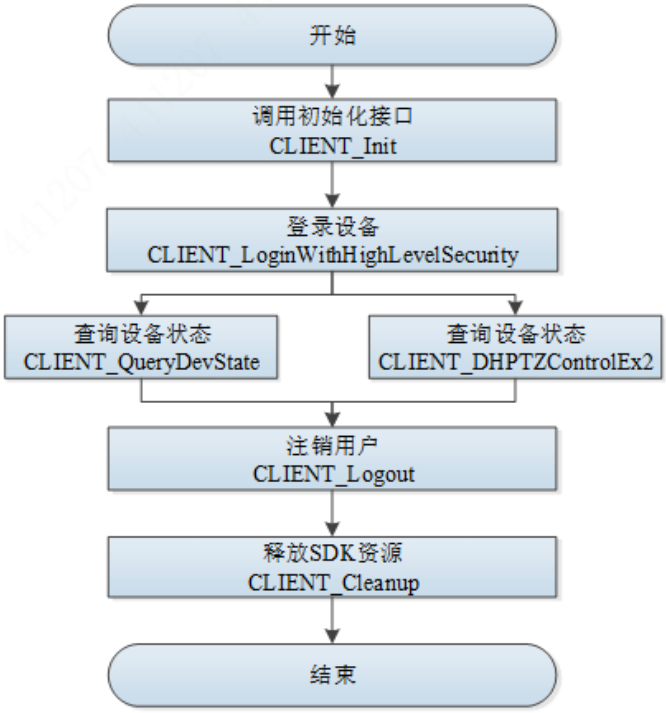
2.8.1.1 简介

获取、增加/修改、删除预置点

2.8.1.2 接口总览

接口	说明
CLIENT_QueryDevState	获取预置点列表。 参数 nType 为 DH_DEVSTATE_PTZ_PRESET_LIST。 取 NET_PTZ_PRESET 结构体中 nIndex，szName 字段，其他字段忽略
CLIENT_DHPTZControlEx2	增加、修改预置点。 参数 dwPTZCommand 为 DH_PTZ_POINT_SET_CONTROL。
CLIENT_DHPTZControlEx2	删除预置点。 参数 dwPTZCommand 为 DH_PTZ_POINT_DEL_CONTROL。

2.8.1.3 流程说明



步骤1　调用 CLIENT_Init 完成 SDK 初始化流程。
步骤2　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。
步骤3　调用 CLIENT_QueryDevState 查询设备状态。
步骤4　业务使用完后，调用 CLIENT_Logout 登出设备。
步骤5　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

2.8.1.4 示例代码

```
//获取当前云台的预置点列表
void QueryPtzPresetList()
{
    int RetLen = 0;
    NET_PTZ_PRESET_LIST Info;
    Info.dwSize = sizeof(NET_PTZ_PRESET_LIST);  //必填
    Info.dwMaxPresetNum = 20;  //必填
    Info.pstuPtzPorsetList = new NET_PTZ_PRESET[Info.dwMaxPresetNum]; //必填

    BOOL QueryDevStateReturn = CLIENT_QueryDevState(lLoginHandle, DH_DEVSTATE_PTZ_PRESET_LIST, (char*)&Info, sizeof(Info), &RetLen, 5000);
    if(QueryDevStateReturn)
    {
        printf("CLIENT_QueryDevState Success 预置点个数=%d;\n", Info.dwRetPresetNum);
    }
    else
    {
        printf("CLIENT_QueryDevState Failed ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}


//增加、修改预置点
void PTZControlPointSet()
{
    bool DHPTZControlEx2Return = CLIENT_DHPTZControlEx2(lLoginHandle, 0, DH_PTZ_POINT_SET_CONTROL, NULL, 8, NULL, FALSE, "预置点 8");
    if(DHPTZControlEx2Return)
    {
        printf("CLIENT_DHPTZControlEx2 Success\n");
    }
    else
    {
        printf("CLIENT_DHPTZControlEx2 Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}


//删除预置点
void PTZControlPointDel()
{
    bool DHPTZControlEx2Return = CLIENT_DHPTZControlEx2(lLoginHandle, 0, DH_PTZ_POINT_DEL_CONTROL, NULL, 8, NULL, FALSE, NULL);
    if(DHPTZControlEx2Return)
    {
        printf("CLIENT_DHPTZControlEx2 Success\n");
```

```
    }
    else
    {
        printf("CLIENT_DHPTZControlEx2 Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}
```

2.8.2 云台绝对位置

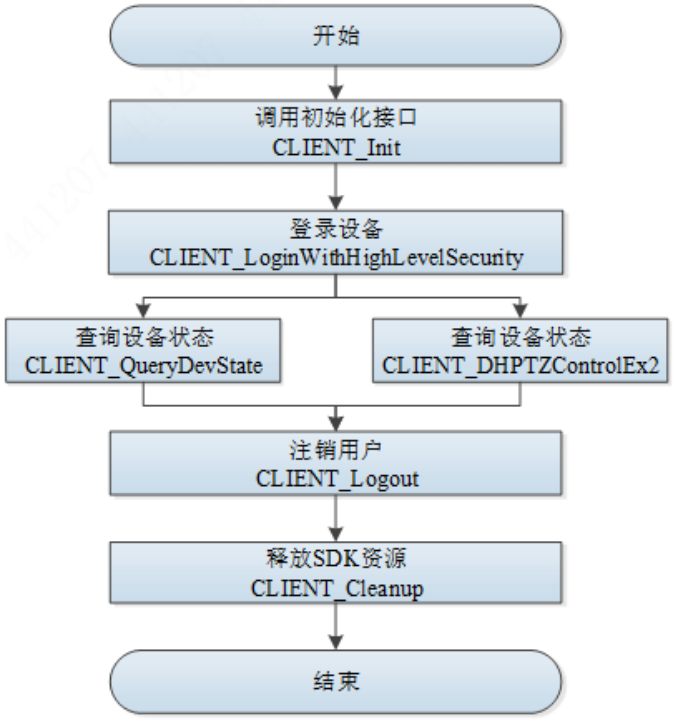
2.8.2.1 简介

获取、设置云台绝对位置

2.8.2.2 接口总览

接口	说明
CLIENT_QueryDevState	获取云台绝对位置。 参数 nType 为 DH_DEVSTATE_PTZ_LOCATION。 取 DH_PTZ_LOCATION_INFO 结构体中 nPTZPan[0,3600]、nPTZTilt[-1800,1800]、nPTZZoom[0,128] 字段，其他字段忽略
CLIENT_DHPTZControlEx2	设置云台绝对位置。 参数 dwPTZCommand 为 DH_EXTPTZ_MOVE_ABSOLUTELY。

2.8.2.3 流程说明



2.8.2.4 示例代码

```
//获取云台位置
void QueryPtzLocationPTZ()
{
    int RetLen = 0;
    DH_PTZ_LOCATION_INFO Info;
```

```
    Info.nChannelID = 0;   //必填

    BOOL QueryDevStateReturn = CLIENT_QueryDevState(lLoginHandle,
DH_DEVSTATE_PTZ_LOCATION, (char*)&Info, sizeof(Info), &RetLen, 5000);
    if(QueryDevStateReturn)
    {
        printf("CLIENT_QueryDevState Success; P 值=%d、T 值=%d、变倍值=%d
\n",Info.nPTZPan,Info.nPTZTilt,Info.nPTZZoom);
    }
    else
    {
        printf("CLIENT_QueryDevState Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}


//设置云台位置
void PTZControlMoveAbsolutely()
{
    PTZ_CONTROL_ABSOLUTELY info;
    info.stuPosition.nPositionX = 500; //P 值,范围[0,3600]
    info.stuPosition.nPositionY = 500; //T 值,范围[-1800,1800]
    info.stuPosition.nZoom = 2;              //变倍值,范围[0,128]
    info.stuSpeed.fPositionX = 0.9;
    info.stuSpeed.fPositionY = 0.9;
    info.stuSpeed.fZoom = 0.9;

    bool DHPTZControlEx2Return = CLIENT_DHPTZControlEx2(lLoginHandle, 0,
DH_EXTPTZ_MOVE_ABSOLUTELY, NULL, NULL, NULL, FALSE, &info);
    if(DHPTZControlEx2Return)
    {
        printf("CLIENT_DHPTZControlEx2 Success\n");
    }
    else
    {
        printf("CLIENT_DHPTZControlEx2 Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}
```

2.8.3 云台持续控制

2.8.3.1 简介

云台八方向、变倍、变焦、光圈控制

2.8.3.2 接口总览

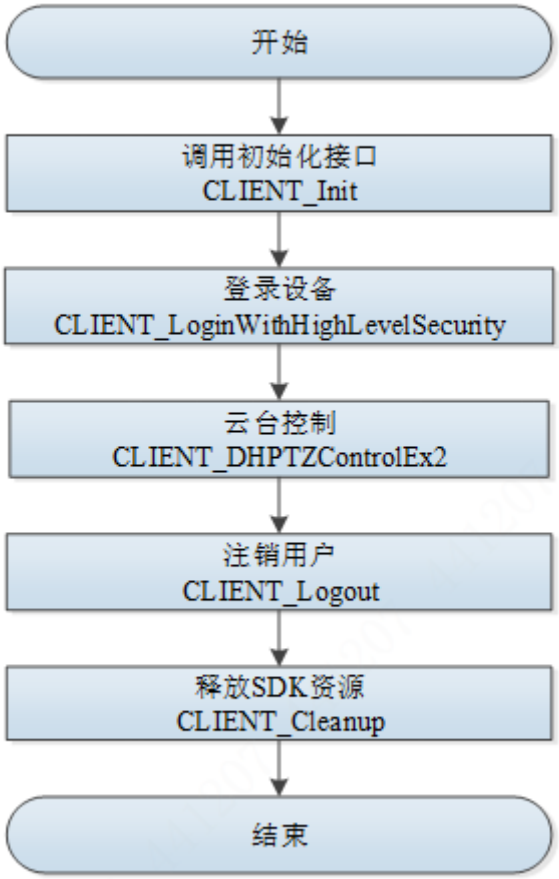
接口	说明
CLIENT_DHPTZControlEx2	参数 1：登录句柄 参数 2：通道号，根据实际填写

接口	说明
	参数 3： 开始 dwPTZCommand 为 DH_EXTPTZ_START 结束 dwPTZCommand 为 DH_EXTPTZ_STOP 参数 4：填写 NULL 参数 5：填写 NULL 参数 6：填写 NULL 参数 7：填写 FALSE 参数 8： 开始 param4 为 NET_IN_PTZ_START_INFO 结束 param4 为 NET_IN_PTZ_STOP_INFO

表2-8 NET_IN_PTZ_START_INFO、NET_IN_PTZ_STOP_INFO 填写说明

szCode	说明	nArg1	nArg2	nArg3	nArg4	备注
"Up"	上	步长(1-8)				
"Down"	下	步长(1-8)				
"Left"	左	步长(1-8)				
"Right"	右	步长(1-8)				
"ZoomWide"	变倍（大）	步长(1-8)				
"ZoomTele"	变倍（小）	步长(1-8)				
"FocusNear"	聚焦（近）	步长(1-8)				
"FocusFar"	聚焦（远）	步长(1-8)				
"IrisLarge"	光圈（变大）	步长(1-8)				
"IrisSmall"	光圈（变小）	步长(1-8)				
"GotoPreset"	转至预置点	预置点值				
"SetPreset"	设置预置点	预置点值				
"ClearPreset"	清除预置点	预置点值				
"Wiper"	雨刷	开 1，关 0				
"StartTour"	开始巡航	巡航线路				
"AutoTour"	自动巡航					
"StopTour"	停止巡航	巡航线路				
"LeftUp"	左上	步长(1-8)	步长(1-8)			
"RightUp"	右上	步长(1-8)	步长(1-8)			
"LeftDown"	左下	步长(1-8)	步长(1-8)			
"RightDown"	右下	步长(1-8)	步长(1-8)			
"AddTour"	加入预置点到巡航	巡航线路	预置点值			
"DelTour"	删除巡航中预置点	巡航线路	预置点值			
"ClearTour"	清除巡航	巡航线路				
"AutoPanOn"	开始水平旋转					
"AutoPanOff"	停止水平旋转					
"SetLeftLimit"	设置左边界					
"SetRightLimit"	设置右边界					
"AutoScanOn"	开始线扫					
"AutoScanOff"	停止线扫					

2.8.3.3 流程说明



2.8.3.4 示例代码

```
//开始云台控制
void PTZControlStart()
{
	NET_IN_PTZ_START_INFO Info;
	Info.dwSize = sizeof(NET_IN_PTZ_START_INFO);  //必填
	std::string szCode = "Left";
	std::strcpy(Info.szCode, szCode.c_str());
	Info.nArg1 = 2;  //必填，步长(1-8)
	int Channel = 0;  //必填，通道号根据实际填写

	bool DHPTZControlEx2Return = CLIENT_DHPTZControlEx2(lLoginHandle, Channel, DH_EXTPTZ_START, NULL, NULL, NULL, FALSE, &Info);
	if(DHPTZControlEx2Return)
	{
		printf("CLIENT_DHPTZControlEx2 Success\n");
	}
	else
	{
		printf("CLIENT_DHPTZControlEx2 Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
	}
}
```

```
//停止云台控制
void PTZControlStart()
{
	NET_IN_PTZ_STOP_INFO Info;
	Info.dwSize = sizeof(NET_IN_PTZ_STOP_INFO);  //必填
	std::string szCode = "Left";
	std::strcpy(Info.szCode, szCode.c_str());
	Info.nArg1 = 2;  //必填，步长(1-8)
	int Channel = 0; //必填，通道号根据实际填写

	bool DHPTZControlEx2Return = CLIENT_DHPTZControlEx2(lLoginHandle, Channel, DH_EXTPTZ_STOP, NULL, NULL, NULL, FALSE, &Info);
	if(DHPTZControlEx2Return)
	{
		printf("CLIENT_DHPTZControlEx2 Success\n");
	}
	else
	{
		printf("CLIENT_DHPTZControlEx2 Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
	}
}
```

2.9 火情预警

2.9.1 简介

热成像火点报警事件的订阅以及报警回调信息的解析

2.9.2 接口总览

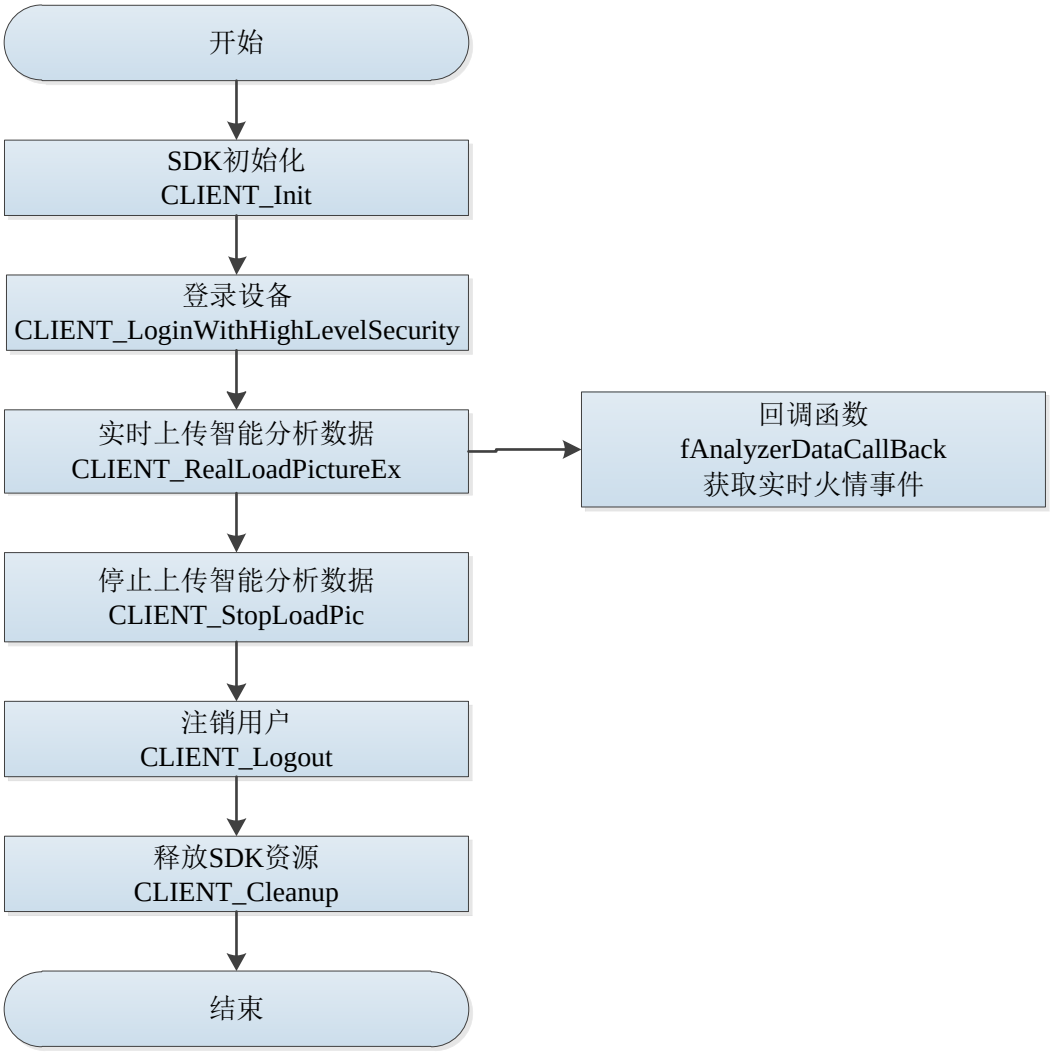
表2-9 火点接口信息

接口	说明
CLIENT_RealLoadPictureEx	订阅火点事件
CLIENT_StopLoadPic	取消订阅火点事件
CLIENT_SetDVRMessCallBack	设置报警回调函数
CLIENT_StartListenEx	订阅火点报警信息
CLIENT_StopListen	取消订阅火点报警信息

2.9.3 流程说明

2.9.3.1 订阅火点事件

图2-10 订阅火点事件流程



流程说明

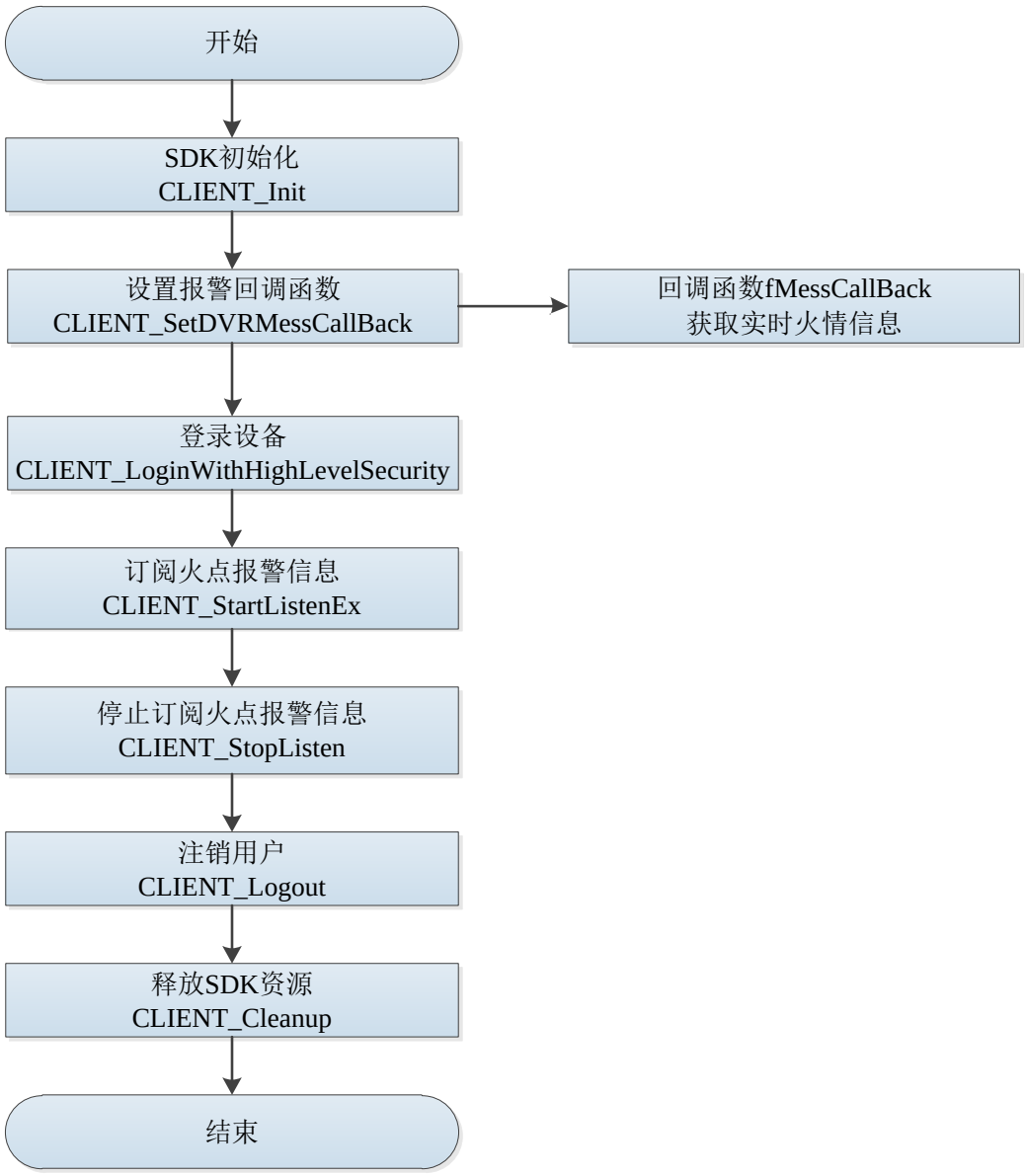
步骤1　完成 SDK 初始化流程。
步骤2　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。
步骤3　调用 CLIENT_RealLoadPictureEx 订阅火点事件，当事件触发时 fAnalyzerDataCallBack 会提示用户。
步骤4　调用 CLIENT_StopLoadPic 取消订阅火点事件。
步骤5　调用 CLIENT_Logout，登出设备。
步骤6　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

订阅火点事件，对应设备端的 FireWarning 事件，事件触发后会联动抓图。火点出现时会发送 start，火点消失时会发送 stop。

2.9.3.2 订阅火点信息

图2-11 订阅火点信息流程



流程说明

步骤1　完成 SDK 初始化流程。

步骤2　调用 CLIENT_SetDVRMessCallBack 设置报警回调函数，当有事件触发时 fMessCallBack 会提示用户。

步骤3　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。

步骤4　调用 CLIENT_StartListenEx 订阅火点信息。

步骤5　调用 CLIENT_StopListen 取消订阅火点信息。

步骤6　调用 CLIENT_Logout，登出设备。

步骤7　SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 订阅火点信息，对应设备端的 FireWarningInfo 事件，该事件在 FireWarning 事件触发后上报，主要携带火点触发的具体信息，包括火点的个数、火点位置坐标、当前设备的 PTZ 值、设备的 GPS 等信息，并且周期性上报，以便及时更新火点信息。
- 由于 FireWarningInfo 是状态更新事件，该事件不能联动抓图录像等常规联动项。

2.9.4 示例代码

```
//订阅火点事件
LDWORD dwUser = 0;
g_lRealLoadHandle = CLIENT_RealLoadPictureEx(g_lLoginHandle, 0, EVENT_IVS_FIREWARNING, TRUE, AnalyzerDataCallBack, dwUser, NULL);
if (0 == g_lRealLoadHandle)
{
    //订阅失败
    return;
}
//停止订阅火点事件
if (0 != g_lRealLoadHandle)
{
    if (FALSE == CLIENT_StopLoadPic(g_lRealLoadHandle))
    {
        //停止订阅失败
    }
    else
    {
        g_lRealLoadHandle = 0;
    }
}
//订阅火点信息
CLIENT_SetDVRMessCallBack(MessCallBack , NULL);
if( CLIENT_StartListenEx(g_lLoginHandle))
{
    g_bStartListenFlag = TRUE;
    printf("Listen Success!\nJust Wait Event....\n");
}
else
{
    //订阅火点信息失败
}
//停止订阅火点信息
if (g_bStartListenFlag)
{
    if (!CLIENT_StopListen(g_lLoginHandle))
    {
        //停止订阅失败
    }
    else
    {
        g_bStartListenFlag = FALSE;
    }
}
```

2.10 冷热点获取

2.10.1 简介

测温型设备支持。

2.10.2 接口总览

接口	说明
CLIENT_RadiometryGetCurrentHotColdSpotInfo	获取热成像当前冷（最低的温度）、热（最高的温度）点信息。 使用此接口需满足以下条件： 1、 热成像测温型设备 2、 开启 web 界面冷热点使能

2.10.3 流程说明



2.10.4 示例代码

```
//获取热成像当前冷（最低的温度）、热（最高的温度）点信息
void GetCurrentHotColdSpotInfo()
{
    NET_IN_RADIOMETRY_CURRENTHOTCOLDSPOT_INFO inInfo;
    inInfo.dwSize = sizeof(NET_IN_RADIOMETRY_CURRENTHOTCOLDSPOT_INFO);    //必填
    inInfo.nChannel = 1;    //必填，填写热成像通道号

    NET_OUT_RADIOMETRY_CURRENTHOTCOLDSPOT_INFO outInfo;
    outInfo.dwSize = sizeof(NET_OUT_RADIOMETRY_CURRENTHOTCOLDSPOT_INFO); //必填
```

```
    BOOL Return = CLIENT_RadiometryGetCurrentHotColdSpotInfo(lLoginHandle, &inInfo, &outInfo, 5000);
    if (Return)
    {
        printf("GetCurrentHotColdSpotInfo Success; 最高温度=%f\n", outInfo.stuCurrentHotColdSpotInfo.fHotSpotValue);
    }
    else
    {
        printf("GetCurrentHotColdSpotInfo Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}
```

2.11 随机区域温度信息

2.11.1 简介

测温型设备支持。

2.11.2 接口总览

接口	说明
CLIENT_RadiometryGetRandomRegionTemper	获取测温区域的参数值。 此接口仅测温型设备支持。

2.11.3 流程说明



2.11.4 示例代码

```
//获取随机区域的温度数据
void GetRandomRegionTemper()
{
    NET_IN_RADIOMETRY_RANDOM_REGION_TEMPER inInfo;
    inInfo.dwSize = sizeof(NET_IN_RADIOMETRY_RANDOM_REGION_TEMPER);  //必填
    inInfo.nChannel = 0;                    //必填，填写热成像通道号
    inInfo.nPointNum = 4;                   //必填，图形顶点数
    //以下图形为矩形
    inInfo.stuPolygon[0].nx = 10;               //必填
    inInfo.stuPolygon[0].ny = 10;               //必填
    inInfo.stuPolygon[1].nx = 8000;         //必填
    inInfo.stuPolygon[1].ny = 10;               //必填
    inInfo.stuPolygon[2].nx = 8000;         //必填
    inInfo.stuPolygon[2].ny = 8000;         //必填
    inInfo.stuPolygon[3].nx = 10;               //必填
    inInfo.stuPolygon[3].ny = 8000;         //必填


    NET_OUT_RADIOMETRY_RANDOM_REGION_TEMPER outInfo;
    outInfo.dwSize = sizeof(NET_OUT_RADIOMETRY_RANDOM_REGION_TEMPER);  //必
填

    bool Return = CLIENT_RadiometryGetRandomRegionTemper(lLoginHandle, &inInfo,
&outInfo, 15000);
    if (Return)
    {
        printf("GetRandomRegionTemper Success 随机区域最高温=%d\n",
outInfo.stuRegionTempInfo.nTemperMax);
    }
    else
    {
        printf("GetRandomRegionTemper Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}
```

2.12 测温设置

2.12.1 规则设置

获取、添加、删除测温规则。

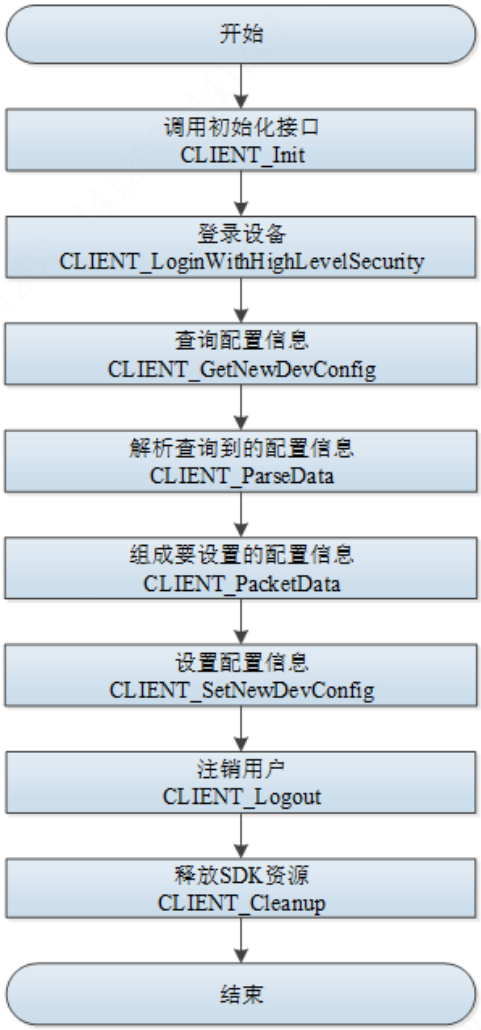
2.12.1.1 简介

测温型设备支持。

2.12.1.2 接口总览

接口	说明
CLIENT_GetNewDevConfig	查询测温规则。 参数 szCommand 为 CFG_CMD_THERMOMETRY_RULE。
CLIENT_ParseData	解析查询到的测温规则。 参数 szCommand 为 CFG_CMD_THERMOMETRY_RULE。
CLIENT_PacketData	组成要设置的测温规则。 参数 szCommand 为 CFG_CMD_THERMOMETRY_RULE。
CLIENT_SetNewDevConfig	设置测温规则。 参数 szCommand 为 CFG_CMD_THERMOMETRY_RULE。

2.12.1.3 流程说明



2.12.1.4 示例代码

```
void ThermometryRule()
{
    //查询测温规则
    int Channel = -1;                    //必填，通道号根据实际填写
    char szJsonBuf[40*512] = {0};
    int nBufferLen = 40*512;
    int nerror = 0;
    bool    GetNewDevConfigReturn    =    CLIENT_GetNewDevConfig(lLoginHandle,
```

```
CFG_CMD_THERMOMETRY_RULE, Channel, szJsonBuf, nBufferLen, &nerror, 5000);
    if(GetNewDevConfigReturn)
    {
        printf("CLIENT_GetNewDevConfig Success\n");

        //解析查询到的测温规则
        CFG_RADIOMETRY_RULE_INFO* RuleInfo = new CFG_RADIOMETRY_RULE_INFO;
        memset(RuleInfo, 0, sizeof(CFG_RADIOMETRY_RULE_INFO));
        bool ParseDataReturn = CLIENT_ParseData(CFG_CMD_THERMOMETRY_RULE, szJsonBuf, RuleInfo, sizeof(CFG_RADIOMETRY_RULE_INFO), NULL);
        if(ParseDataReturn)
        {
            printf("CLIENT_ParseData Success; 规则个数=%d、测温使能=%d、预置点编号=%d\n",RuleInfo->nCount, RuleInfo->stRule[0].bEnable, RuleInfo->stRule[0].nPresetId);

            //组成要设置的测温规则
            RuleInfo->stRule[0].bEnable = 0;
            char szJsonBuf2[40*512] = {0};
            int nBufferLen2 = 40*512;
            bool PacketDataReturn = CLIENT_PacketData(CFG_CMD_THERMOMETRY_RULE, RuleInfo, sizeof(CFG_RADIOMETRY_RULE_INFO), szJsonBuf2, nBufferLen2);
            if(PacketDataReturn)
            {
                printf("CLIENT_PacketData Success\n");

                int Restart = 0;  //是否需要重启设备，1 表示需要重启，0 表示不需要重启
                bool SetNewDevConfigReturn = CLIENT_SetNewDevConfig(lLoginHandle, CFG_CMD_THERMOMETRY_RULE, Channel, szJsonBuf2, nBufferLen2, &nerror, &Restart, 5000);
                if(SetNewDevConfigReturn)
                {
                    printf("CLIENT_SetNewDevConfig Success \n");
                }
                else
                {
                    printf("CLIENT_SetNewDevConfig Fail ErrorCode: 0x%x; Error=%d\n", CLIENT_GetLastError(), nerror);
                }
            }
            else
            {
                printf("CLIENT_PacketData Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
            }
        }
        else
        {
            printf("CLIENT_ParseData Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
```

```
        }
    }
    else
    {
        printf("CLIENT_GetNewDevConfig Fail ErrorCode: 0x%x; Error=%d\n",
CLIENT_GetLastError(), nerror);
    }
}
```

2.12.2 全局设置

获取、修改测温全局设置。

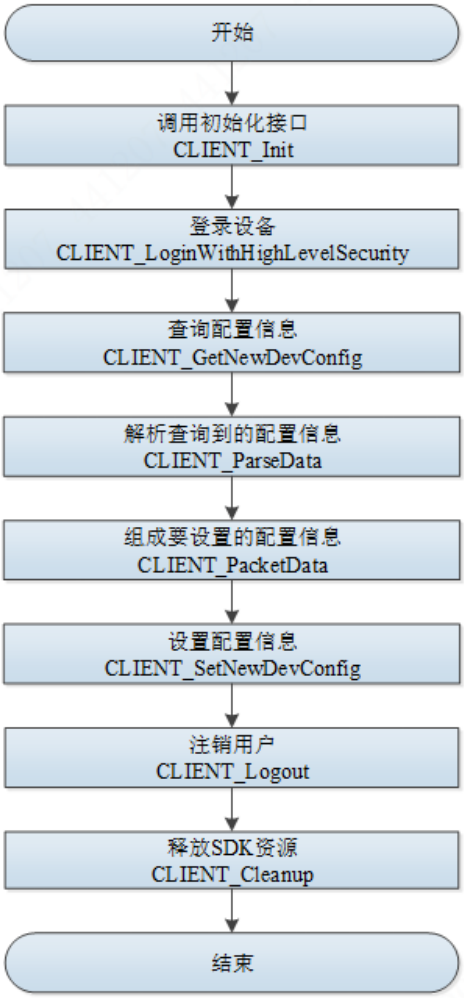
2.12.2.1 简介

测温型设备支持。

2.12.2.2 接口总览

接口	说明
CLIENT_GetNewDevConfig	查询测温全局配置。 参数 szCommand 为 CFG_CMD_THERMOMETRY。
CLIENT_ParseData	解析查询到的测温全局配置。 参数 szCommand 为 CFG_CMD_THERMOMETRY。
CLIENT_PacketData	组成要设置的测温全局配置。 参数 szCommand 为 CFG_CMD_THERMOMETRY。
CLIENT_SetNewDevConfig	设置测温全局配置。 参数 szCommand 为 CFG_CMD_THERMOMETRY。

2.12.2.3 流程说明



2.12.2.4 示例代码

```
void Thermometry()
{
    //查询测温全局配置
    int Channel = -1;                    //必填，通道号根据实际填写
    char szJsonBuf[40*512] = {0};
    int nBufferLen = 40*512;
    int nerror = 0;
    bool GetNewDevConfigReturn = CLIENT_GetNewDevConfig(lLoginHandle, CFG_CMD_THERMOMETRY, Channel, szJsonBuf, nBufferLen, &nerror, 5000);
    if(GetNewDevConfigReturn)
    {
        printf("CLIENT_GetNewDevConfig Success\n");

        //解析查询到的测温全局配置
        CFG_THERMOMETRY_INFO* ThermometryInfo = new CFG_THERMOMETRY_INFO;
        memset(ThermometryInfo, 0, sizeof(CFG_THERMOMETRY_INFO));
        bool ParseDataReturn = CLIENT_ParseData(CFG_CMD_THERMOMETRY, szJsonBuf, ThermometryInfo, sizeof(CFG_THERMOMETRY_INFO), NULL);
        if(ParseDataReturn)
```

```
        {
            printf("CLIENT_ParseData Success; 相 对 湿 度 =%d \n",ThermometryInfo->nRelativeHumidity);

            //组成要设置的测温全局配置
            ThermometryInfo->nRelativeHumidity = 88;
            char szJsonBuf2[40*512] = {0};
            int nBufferLen2 = 40*512;
            bool PacketDataReturn = CLIENT_PacketData(CFG_CMD_THERMOMETRY, ThermometryInfo, sizeof(CFG_THERMOMETRY_INFO), szJsonBuf2, nBufferLen2);
            if(PacketDataReturn)
            {
                printf("CLIENT_PacketData Success\n");

                //设置测温全局配置
                int Restart = 0;  //是否需要重启设备，1 表示需要重启，0 表示不需要重启
                bool SetNewDevConfigReturn = CLIENT_SetNewDevConfig(lLoginHandle, CFG_CMD_THERMOMETRY, Channel, szJsonBuf2, nBufferLen2, &nerror, &Restart, 5000);
                if(SetNewDevConfigReturn)
                {
                    printf("CLIENT_SetNewDevConfig Success \n");
                }
                else
                {
                    printf("CLIENT_SetNewDevConfig Fail ErrorCode: 0x%x; Error=%d\n", CLIENT_GetLastError(), nerror);
                }
            }
            else
            {
                printf("CLIENT_PacketData Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
            }
        }
        else
        {
            printf("CLIENT_ParseData Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
        }
    }
    else
    {
        printf("CLIENT_GetNewDevConfig Fail ErrorCode: 0x%x; Error=%d\n", CLIENT_GetLastError(), nerror);
    }
}
```

2.12.3 温度异常报警

设置测温规则后，根据设置的测温规则，上报温度异常报警。

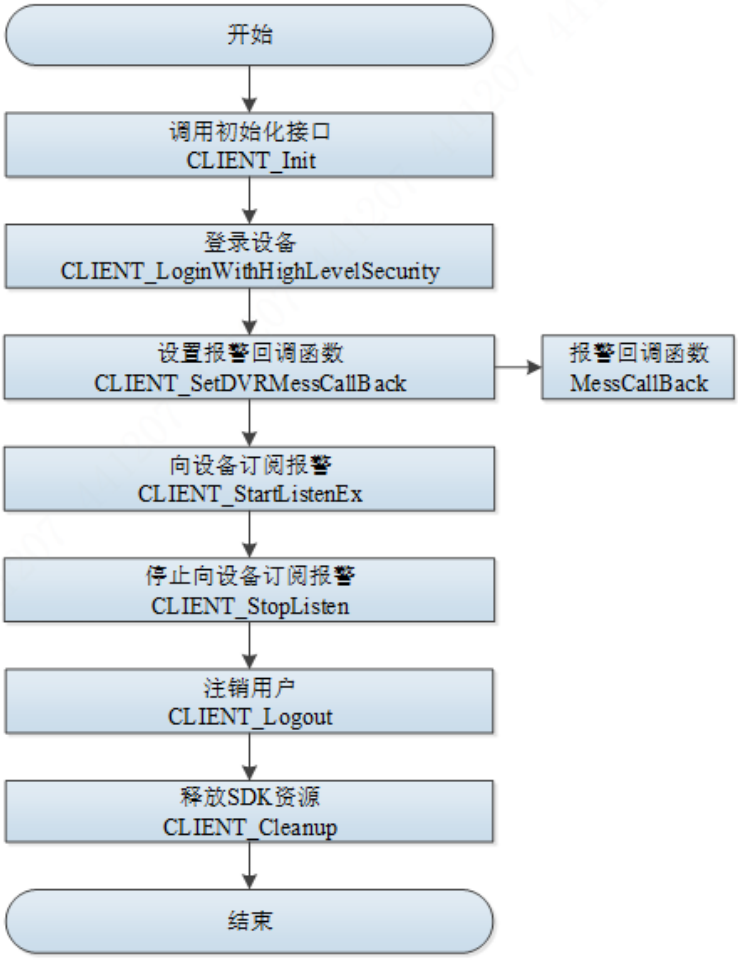
2.12.3.1 简介

测温型设备支持。

2.12.3.2 接口总览

接口	说明
CLIENT_SetDVRMessCallBack	设置报警回调函数。 参数 fMessCallBack 中 lCommand 为 DH_ALARM_HEATIMG_TEMPER。参数 fMessCallBack 中 pBuf 为 ALARM_HEATIMG_TEMPER_INFO。
CLIENT_StartListenEx	向设备订阅报警。
CLIENT_StopListen	停止订阅报警。

2.12.3.3 流程说明



流程说明

步骤1　调用 CLIENT_Init 函数，完成 SDK 初始化流程。

步骤2　初始化成功后，调用 CLIENT_LoginWithHighLevelSecurity 登录设备。

步骤3　调用 CLIENT_SetDVRMessCallBack 函数，设置报警回调函数，该接口需在报警订阅之前调用。

步骤4　调用 CLIENT_StartListenEx 函数，向设备订阅报警。订阅成功后，设备上报的报警事件通过 CLIENT_SetDVRMessCallBack 函数设置的回调函数通知用户。

步骤5　报警上报功能使用完毕后，调用 CLIENT_StopListen 函数停止向设备订阅报警。

步骤6　调用 CLIENT_Logout 函数登出设备。

步骤7　SDK 功能使用完后，调用 CLIENT_Cleanup 函数释放 SDK 资源。

注意事项

如果之前能上报的报警突然不再上报了，则需排查下设备是否断线。

2.12.3.4 示例代码

```
//消息回调处理函数
BOOL CALLBACK MessCallBack(LONG lCommand, LLONG lLoginID, char *pBuf, DWORD dwBufLen, char *pchDVRIP, LONG nDVRPort, LDWORD dwUser)
{
    //过滤热成像测温点温度异常报警
    if (DH_ALARM_HEATIMG_TEMPER == lCommand)
    {
        ALARM_HEATIMG_TEMPER_INFO* pInfo = (ALARM_HEATIMG_TEMPER_INFO*)pBuf;

        printf("热成像测温点温度异常报警 ALARM; 温度异常点名称=%s、报警项编号=%d、报警结果值=%d、报警条件=%d \n",
            pInfo->szName, pInfo->nAlarmId, pInfo->nResult, pInfo->nAlarmContion);
    }
    return TRUE;
}

//开始
void StartListenEx()
{
    //设置报警回调函数
    CLIENT_SetDVRMessCallBack(MessCallBack, NULL);

    //向设备订阅报警
    BOOL StartListenExReturn = CLIENT_StartListenEx(lLoginHandle);
    if (StartListenExReturn)
    {
        printf("CLIENT_StartListenEx Success\n");
    }
    else
    {
        printf("CLIENT_StartListenEx Fail ErrorCode: 0x%x \n", CLIENT_GetLastError());
    }

    printf("Print any key to stop listen\n");
    getchar();

    //停止订阅报警
    BOOL StopListenReturn = CLIENT_StopListen(lLoginHandle);
    if(StopListenReturn)
    {
        printf("CLIENT_StopListen Success\n");
```

```
    }
    else
    {
        printf("CLIENT_StopListen Fail ErrorCode: 0x%x \n", CLIENT_GetLastError());
    }
}
```

2.12.4 获取规则温度信息

2.12.4.1 简介

测温型设备支持。

2.12.4.2 接口总览

接口	说明
CLIENT_RadiometryGetCurTemperAll	获取测温规则的温度信息。

2.12.4.3 流程说明



2.12.4.4 示例代码

```
//获取测温规则的温度信息
void GetCurTemperAll()
{
    NET_IN_RADIOMETRY_GET_CUR_TEMPER_ALL_INFO inInfo;
    inInfo.dwSize = sizeof(NET_IN_RADIOMETRY_GET_CUR_TEMPER_ALL_INFO);    //必填
    inInfo.nChannel = 1;                                                  //必填，填写热成像通道号

    NET_OUT_RADIOMETRY_GET_CUR_TEMPER_ALL_INFO outInfo;
    outInfo.dwSize    =    sizeof(NET_OUT_RADIOMETRY_GET_CUR_TEMPER_ALL_INFO);
```

```
//必填
    outInfo.nMaxTempInfoNum = 20;
//必填
    outInfo.pstTempInfo = new NET_RADIOMETRY_TEMP_INFO[outInfo.nMaxTempInfoNum]; //必填
    memset(outInfo.pstTempInfo, 0x00, sizeof(NET_RADIOMETRY_TEMP_INFO)*outInfo.nMaxTempInfoNum);

    BOOL Return = CLIENT_RadiometryGetCurTemperAll(lLoginHandle, &inInfo, &outInfo, 5000);
    if (Return)
    {
        printf("GetCurTemperAll Success; 实际返回的规则温度信息数量=%d\n", outInfo.nRetTempInfoNum);
    }
    else
    {
        printf("GetCurTemperAll Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}
```

2.12.5 主动上报规则温度信息

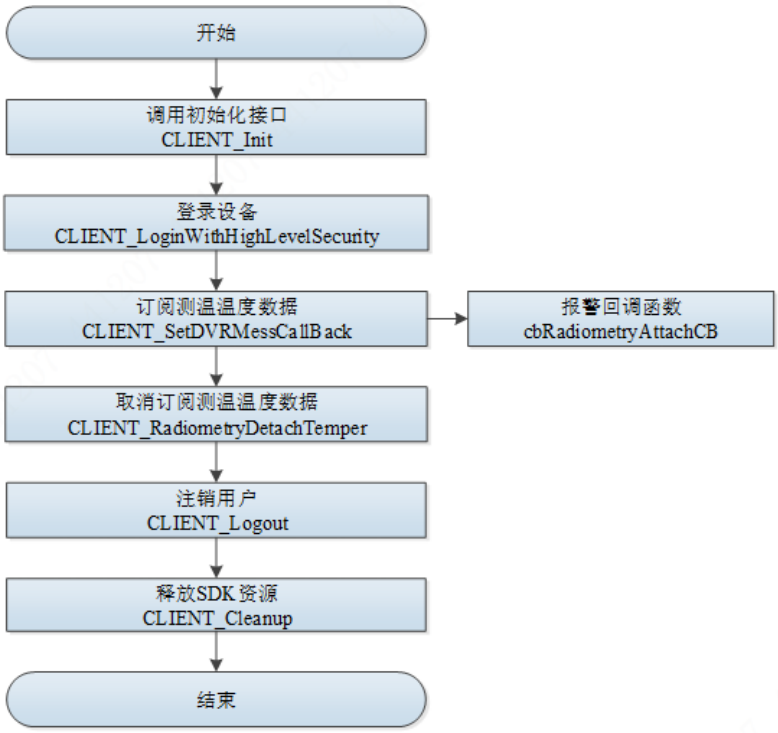
2.12.5.1 简介

测温型设备支持。

2.12.5.2 接口总览

接口	说明
CLIENT_RadiometryAttachTemper	订阅测温温度数据。
CLIENT_RadiometryDetachTemper	取消订阅测温温度数据。

2.12.5.3 流程说明



2.12.5.4 示例代码

```
LLONG m_lAttachhandle = 0;

void CALL_METHOD cbRadiometryAttachCB(LLONG lAttachTemperHandle,
NET_RADIOMETRY_TEMPER_DATA* pBuf, int nBufLen, LDWORD dwUser)
{
    for(int i = 0; i < pBuf->nRadiometryTemperNum; i++)
    {
        printf("%d_%d_%d_%d_%d_%d_fTemperAve:%f \n",
            pBuf->stuRadiometryTemperInfo[i].stuTime.dwYear,
            pBuf->stuRadiometryTemperInfo[i].stuTime.dwMonth,
            pBuf->stuRadiometryTemperInfo[i].stuTime.dwDay,
            pBuf->stuRadiometryTemperInfo[i].stuTime.dwHour,
            pBuf->stuRadiometryTemperInfo[i].stuTime.dwMinute,
            pBuf->stuRadiometryTemperInfo[i].stuTime.dwSecond,
            pBuf->stuRadiometryTemperInfo[i].stuQueryTemperInfo.fTemperAve);
    }
}

void AttachTemper()
{

    NET_IN_RADIOMETRY_ATTACH_TEMPER inInfo;
    inInfo.dwSize = sizeof(NET_IN_RADIOMETRY_ATTACH_TEMPER);

    std::cin >> inInfo.nChannel;
    inInfo.nChannel = 1;
```

```
    inInfo.cbNotify = cbRadiometryAttachCB;

    NET_OUT_RADIOMETRY_ATTACH_TEMPER outInfo;
    outInfo.dwSize = sizeof(NET_OUT_RADIOMETRY_ATTACH_TEMPER);

    bool m_lAttachhandle = CLIENT_RadiometryAttachTemper(lLoginHandle, &inInfo, &outInfo, 5000);
    if (m_lAttachhandle > 0)
    {
        printf("CLIENT_RadiometryAttachTemper Success\n");
    }
    else
    {
        printf("CLIENT_RadiometryAttachTemper Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }

}

void DetachTemper()
{
    BOOL Return = CLIENT_RadiometryDetachTemper(m_lAttachhandle);
    if (Return)
    {
        printf("CLIENT_RadiometryDetachTemper Success\n");
    }
    else
    {
        printf("CLIENT_RadiometryDetachTemper Fail ErrorCode: 0x%x\n", CLIENT_GetLastError());
    }
}
```

第 3 章 接口说明

3.1 SDK 初始化

3.1.1 SDK 初始化 CLIENT_Init

选项	说明	
描述	对整个 SDK 进行初始化	
函数	BOOL CLIENT_Init(fDisConnect cbDisConnect, LDWORD dwUser);	
参数	[in]cbDisConnect	断线回调函数
	[in]dwUser	断线回调函数的用户参数
返回值	成功返回 TRUE，失败返回 FALSE	
说明	● 调用网络 SDK 其他函数的前提 ● 回调函数设置成 NULL 时，设备断线后不会回调给用户	

3.1.2 SDK 清理 CLIENT_Cleanup

选项	说明
描述	清理 SDK
函数	void CLIENT_Cleanup()
参数	无
返回值	无
说明	SDK 清理接口，在结束前最后调用

3.1.3 设置断线重连回调函数 CLIENT_SetAutoReconnect

选项	说明	
描述	设置自动重连回调函数	
函数	void CLIENT_SetAutoReconnect(fHaveReConnect cbAutoConnect, LDWORD dwUser);	
参数	[in]cbAutoConnect	断线重连回调函数
	[in]dwUser	断线重连回调函数的用户参数
返回值	无	
说明	设置断线重连回调接口。如果回调函数设置为 NULL，则不自动重连	

3.1.4 设置网络参数 CLIENT_SetNetworkParam

选项	说明	
描述	设置网络环境相关参数	
函数	void CLIENT_SetNetworkParam(NET_PARAM *pNetParam);	
参数	[in]pNetParam	网络延迟、重连次数、缓存大小等参数
返回值	无	

选项	说明
说明	可根据实际网络环境，调整参数

3.2 设备登录

3.2.1 高安全级别登录 CLIENT_LoginWithHighLevelSecurity

选项	说明		
描述	用户登录设备		
函数	LLONG CLIENT_LoginWithHighLevelSecurity (NET_IN_LOGIN_WITH_HIGHLEVEL_SECURITY* pstInParam, NET_OUT_LOGIN_WITH_HIGHLEVEL_SECURITY* pstOutParam);		
参数	[in] pstInParam	[in] dwSize	结构体大小
		[in] szIP	设备 IP
		[in] nPort	设备端口
		[in] szUserName	用户名
		[in] szPassword	密码
		[in] emSpecCap	登录类别
		[in] pCapParam	登录类别参数
	[out] pstOutParam	[in]dwSize	结构体大小
		[out] stuDeviceInfo	设备信息
		[out] nError	失败的错误码
返回值	成功返回设备 ID，失败返回 0。 登录成功之后对设备的操作都可以通过此值（设备 ID）配合 SDK 接口实现。		
说明	高安全级别登录接口。 **说明** CLIENT_LoginEx2 仍然可以使用，但存在安全风险。所以强烈推荐使用最新接口 CLIENT_LoginWithHighLevelSecurity 登录设备。		

参数 error 的错误码及含义说明，请参见表 3-1。

表3-1 参数 error 的错误码及含义

error 的错误码	对应的含义
1	密码不正确
2	用户名不存在
3	登录超时
4	账号已登录
5	账号已被锁定
6	账号被列为黑名单
7	资源不足，设备系统忙
8	子连接失败
9	主连接失败
10	超过最大用户连接数
11	缺少 avnetsdk 或 avnetsdk 的依赖库
12	设备未插入 U 盘或 U 盘信息错误
13	客户端 IP 地址没有登录权限

3.2.2 用户登出设备 CLIENT_Logout

选项	说明	
描述	用户登出设备	
函数	BOOL CLIENT_Logout(LLONG lLoginID);	
参数	[in]lLoginID	CLIENT_LoginWithHighLevelSecurity 的返回值
返回值	● 成功返回 TRUE ● 失败返回 FALSE	
说明	无	

3.3 实时预览

3.3.1 打开预览 CLIENT_RealPlayEx

选项	说明	
描述	打开实时预览	
函数	LLONG CLIENT_RealPlayEx(LLONG lLoginID, int nChannelID, HWND hWnd, DH_RealPlayType rType);	
参数	[in]lLoginID	CLIENT_LoginWithHighLevelSecurity 的返回值
	[in]nChannelID	视频通道号，从 0 开始递增的整数
	[in]hWnd	窗口句柄，仅在 Windows 系统下有效
	[in]rType	预览类型
返回值	● 成功返回非 0 ● 失败返回 0	
说明	在 Windows 环境下： ● hWnd 为有效值时，在对应窗口显示画面 ● hWnd 为 NULL 时，表示取流方式，通过设置回调函数来获取视频数据，交由用户处理	

预览类型及含义请参见表 3-2。

表3-2 预览类型说明

预览类型	含义
DH_RType_Realplay	实时预览
DH_RType_Multiplay	多画面预览
DH_RType_Realplay_0	实时预览-主码流，等同于 DH_RType_Realplay
DH_RType_Realplay_1	实时预览-从码流 1
DH_RType_Realplay_2	实时预览-从码流 2
DH_RType_Realplay_3	实时预览-从码流 3
DH_RType_Multiplay_1	多画面预览－1 画面
DH_RType_Multiplay_4	多画面预览－4 画面
DH_RType_Multiplay_8	多画面预览－8 画面
DH_RType_Multiplay_9	多画面预览－9 画面
DH_RType_Multiplay_16	多画面预览－16 画面
DH_RType_Multiplay_6	多画面预览－6 画面

预览类型	含义
DH_RType_Multiplay_12	多画面预览－12 画面
DH_RType_Multiplay_25	多画面预览－25 画面
DH_RType_Multiplay_36	多画面预览－36 画面

3.3.2 关闭预览 CLIENT_StopRealPlayEx

选项	说明	
描述	关闭实时预览	
函数	BOOL CLIENT_StopRealPlayEx(LLONG lRealHandle);	
参数	[in]lRealHandle	CLIENT_RealPlayEx 的返回值
返回值	● 成功返回 TRUE ● 失败返回 FALSE	
说明	无	

3.3.3 保存预览数据 CLIENT_SaveRealData

选项	说明	
描述	保存实时预览数据为文件	
函数	BOOL CLIENT_SaveRealData(LLONG lRealHandle, const char *pchFileName);	
参数	[in]lRealHandle	CLIENT_RealPlayEx 的返回值
	[in]pchFileName	需要保存的文件路径
返回值	● 成功返回 TRUE ● 失败返回 FALSE	
说明	无	

3.3.4 停止保存预览数据 CLIENT_StopSaveRealData

选项	说明	
描述	停止保存实时预览数据为文件	
函数	BOOL CLIENT_StopSaveRealData(LLONG lRealHandle);	
参数	[in]lRealHandle	CLIENT_RealPlayEx 的返回值
返回值	● 成功返回 TRUE ● 失败返回 FALSE	
说明	无	

3.3.5 设置预览数据回调 CLIENT_SetRealDataCallBackEx2

选项	说明
描述	设置实时预览数据回调

选项	说明	
函数	BOOL CLIENT_SetRealDataCallBackEx2(LLONG lRealHandle, fRealDataCallBackEx2 cbRealData, LDWORD dwUser, DWORD dwFlag);	
参数	[in]lRealHandle	CLIENT_RealPlayEx 的返回值
	[in]cbRealData	预览数据流回调函数
	[in]dwUser	预览数据流回调函数的参数
	[in]dwFlag	回调中预览数据的类型，EM_REALDATA_FLAG 类型，支持或运算
返回值	● 成功返回 TRUE ● 失败返回 FALSE	
说明	无	

回调中预览数据的类型请参见表 3-3。

表3-3 dwFlag 类型及含义

dwFlag	含义
REALDATA_FLAG_RAW_DATA	原始数据标志
REALDATA_FLAG_DATA_WITH_FRAME_INFO	带有帧信息的数据标志
REALDATA_FLAG_YUV_DATA	YUV 数据标志
REALDATA_FLAG_PCM_AUDIO_DATA	PCM 音频数据标志

3.4 视频抓图

3.4.1 设备抓图 CLIENT_SnapPictureToFile

选项	说明	
描述	抓图	
函数	BOOL CLIENT_SnapPictureToFile(LLONG lLoginID, const NET_IN_SNAP_PIC_TO_FILE_PARAM* pInParam, NET_OUT_SNAP_PIC_TO_FILE_PARAM* pOutParam, int nWaitTime);	
参数	[in]lLoginID	CLIENT_LoginWithHighLevelSecurity 的返回值
	[in]pInParam	输入参数
	[out]pOutParam	输出参数
	[in]nWaitTime	超时时间，单位：毫秒
返回值	● 成功返回 TRUE ● 失败返回 FALSE	
说明	● 同步接口，由设备抓取图片，通过网络传给用户 ● 需要设备支持该功能	

3.4.2 抓图 CLIENT_CapturePictureEx

选项	说明
描述	抓图

选项	说明	
函数	BOOL CLIENT_CapturePictureEx(LLONG hPlayHandle, const char *pchPicFileName, NET_CAPTURE_FORMATS eFormat);	
参数	[in]hPlayHandle	CLIENT_RealPlayEx 的返回值
	[in]pchPicFileName	需要保存的文件路径，必须为绝对路径
	[in]eFormat	图片格式
返回值	● 成功返回 TRUE ● 失败返回 FALSE	
说明	● 同步接口，将图片数据直接写成文件 ● 图片从设备发送过来的实时预览数据流中抓取	

3.5 普通报警上报

3.5.1 设置报警回调函数 CLIENT_SetDVRMessCallBack

选项	说明	
描述	设置报警回调函数	
函数	void CLIENT_SetDVRMessCallBack(fMessCallBack cbMessage, LDWORD dwUser);	
参数	[in] cbMessage	● 消息回调函数，可以回调设备的状态，如报警状态 ● 当设置为 0 时表示禁止回调
	[in] dwUser	用户自定义数据
返回值	无	
说明	● 设置设备消息回调函数，用来得到设备当前状态信息，与调用顺序无关，SDK 默认不回调 ● 此回调函数 fMessCallBack 必须先调用报警消息订阅接口 CLIENT_StartListenEx 才生效	

3.5.2 订阅报警 CLIENT_StartListenEx

选项	说明	
描述	订阅报警	
函数	BOOL CLIENT_StartListenEx(LLONG lLoginID);	
参数	[in] lLoginID	CLIENT_LoginEx2 返回值
返回值	成功返回 TRUE，失败返回 FALSE	
说明	订阅设备消息，得到的消息从 CLIENT_SetDVRMessCallBack 的设置值回调出来	

普通报警事件类型说明请参见表 3-4。

表3-4 普通事件类型说明

dwAlarmType 宏定义	宏定义值	含义	回调 pAlarmInfo 对应结构体
DH_ALARM_ALARM_EX	0x2101	外部报警	DWORD
DH_MOTION_ALARM_EX	0x2102	动态检测	DWORD
DH_ALARM_ALARM_EX2	0x2175	本地报警事件	ALARM_ALARM_INFO_EX2

dwAlarmType 宏定义	宏定义值	含义	回调 pAlarmInfo 对应结构体
DH_ALARM_TEMPERATURE	0x2135	温度过高报警	ALARM_TEMPERATURE_INFO
DH_ALARM_STORAGE_FAILURE_EX	0x2163	存储错误报警	ALARM_STORAGE_FAILURE_EX
DH_ALARM_STORAGE_NOT_EXIST	0x3167	存储组不存在事件	ALARM_STORAGE_NOT_EXIST_INFO
DH_ALARM_ALARM_EX2	0x2175	本地报警事件	ALARM_ALARM_INFO_EX2
DH_ALARM_HEATIMG_TEMPER	0x31aa	热成像测温点温度异常报警事件	ALARM_HEATIMG_TEMPER_INFO
DH_ALARM_FIREWARNING	0x31b5	热成像着火点事件	ALARM_FIREWARNING_INFO
DH_ALARM_SMOKE_DETECTION	0X31d5	烟雾报警事件	ALARM_SMOKE_DETECTION_INFO
DH_ALARM_BETWEENRULE_TEMP_DIFF	0x31d6	热成像规则间温差异常报警	ALARM_BETWEENRULE_DIFFTEMPER_INFO
DH_ALARM_HOTSPOT_WARNING	0X31d8	热成像热点异常报警	ALARM_HOTSPOT_WARNING_INFO
DH_ALARM_COLDSPOT_WARNING	0X31d9	热成像冷点异常报警	ALARM_COLDSPOT_WARNING_INFO
DH_ALARM_FIREWARNING_INFO	0X31da	热成像火情信息上报	ALARM_FIREWARNING_INFO_DETAIL
DH_ALARM_FACE_OVERHEATING	0x31db	热成像人体发烧预警（人脸超温报警）	ALARM_FACE_OVERHEATING_INFO

3.5.3 停止订阅报警 CLIENT_StopListen

选项	说明	
描述	停止订阅报警	
函数	BOOL CLIENT_StopListen(LLONG lLoginID);	
参数	[in] lLoginID	CLIENT_LoginEx2 返回值
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.6 智能事件订阅

3.6.1 开始智能事件订阅 CLIENT_RealLoadPictureEx

选项	说明
描述	开始智能事件订阅

选项	说明	
函数	LLONG CALL_METHOD CLIENT_RealLoadPictureEx(LLONG lLoginID, int nChannelID, DWORD dwAlarmType, BOOL bNeedPicFile, fAnalyzer DataCallBack cbAnalyzerData, LDWORD dwUser, void* Reserved);	
参数	[in] lLoginID	登录句柄
	[in] nChannelID	设备通道号
	[in] dwAlarmType	订阅报警事件类型
	[in]bNeedPicFile	是否订阅图片文件
	[in]cbAnalyzerData	智能事件回调函数
	[in] dwUser	用户自定义数据类型
	[in] Reserved	保留字段
返回值	成功返回 LLONG 类型订阅句柄，失败返回 0	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

智能报警事件类型说明请参见表 3-5。

表3-5 智能事件类型说明

dwAlarmType 宏定义	宏定义值	含义	回调 pAlarmInfo 对应结构体
EVENT_IVS_ALL	0x00000001	所有事件	无
EVENT_IVS_CROSSLINEDETECTION	0x00000002	绊线入侵事件	DEV_EVENT_CROSSLINE_INFO
EVENT_IVS_CROSSREGIONDETECTION	0x00000003	区域入侵事件	DEV_EVENT_CROSSREGION_INFO
EVENT_IVS_FIREDETECTION	0x0000000C	火警事件	DEV_EVENT_FIRE_INFO
EVENT_IVS_SMOKEDETECTION	0x0000000D	烟雾报警事件	DEV_EVENT_SMOKE_INFO

3.6.2 停止智能事件订阅 CLIENT_StopLoadPic

选项	说明	
描述	停止智能事件订阅	
函数	BOOL CALL_METHOD CLIENT_StopRealPlayEx(LLONG lRealHandle);	
参数	[in] lRealHandle	智能事件订阅句柄
返回值	BOOL 类型 ● 成功：TRUE ● 失败：FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

3.7 热图

3.7.1 订阅温度分布数据 CLIENT_RadiometryAttach

选项	说明
描述	开启订阅热图数据

选项	说明	
函数	LLONG CLIENT_RadiometryAttach(LLONG lLoginID, const NET_IN_RADIOMETRY_ATTACH* pInParam, NET_OUT_RADIOMETRY_ATTACH* pOutParam, int nWaitTime);	
参数	[in] lLoginID	CLIENT_LoginEx2 的返回值
	[in] pInParam	输入参数
	[in] pOutParam	输出参数
	[in] nWaitTime	超时时间，单位：毫秒
返回值	成功返回非 0，失败返回 0	
说明	无	

3.7.2 取消订阅温度分布数据 CLIENT_RadiometryDetach

选项	说明	
描述	关闭订阅热图数据	
函数	BOOL CLIENT_RadiometryDetach(LLONG lAttachHandle);	
参数	[in] lAttachHandle	CLIENT_ RadiometryDetach 的返回值
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.7.3 获取温度分布数据 CLIENT_RadiometryFetch

选项	说明	
描述	抓取热图数据	
函数	BOOL CLIENT_RadiometryFetch(LLONG lLoginID, const NET_IN_RADIOMETRY_FETCH* pInParam, NET_OUT_RADIOMETRY_FETCH* pOutParam, int nWaitTime);	
参数	[in] lLoginID	CLIENT_LoginEx2 的返回值
	[in] pInParam	输入参数
	[in] pOutParam	输出参数
	[in] nWaitTime	超时时间，单位：毫秒
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.7.4 解析温度分布数据 CLIENT_RadiometryDataParse

选项	说明
描述	解析热图数据
函数	BOOL CLIENT_RadiometryDataParse(const NET_RADIOMETRY_DATA* pBuf, unsigned short* pImg, float* pTemp);

选项	说明	
参数	[in] pBuf	热图数据
	[in out] pImg	解压后的数据，是一张灰度图（传空指针表示不需要此数据）
	[in out] pTemp	每个像素的温度数据（传空指针表示不需要此数据）
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.8 云台控制

3.8.1 查询设备状态 CLIENT_QueryDevState

选项	说明	
描述	查询设备状态	
函数	BOOL CALL_METHOD CLIENT_QueryDevState(LLONG lLoginID, int nType, char *pBuf, int nBufLen, int *pRetLen, int waittime=1000);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
	[in]nType	查询信息类型
	[out] pBuf	用于接收查询返回的数据的缓存，根据查询类型的不同， 返回数据的数据结构也不同
	[in]nBufLen	缓存长度，单位：字节
	[out]pRetLen	实际返回的数据长度，单位：字节
	[in]waittime	查询状态等待时间，默认 1000ms，可根据需要自行设置
返回值	成功返回 TRUE 失败返回 FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

3.8.2 云台控制 CLIENT_DHPTZControlEx2

选项	说明	
描述	云台控制	
函数	BOOL CLIENT_DHPTZControlEx2(LLONG lLoginID, int nChannelID, DWORD dwPTZCommand, LONG lParam1, LONG lParam2, LONG lParam3, BOOL dwStop , void* param4);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值

选项	说明	
	[in]nChannelID	视频通道号，从 0 开始递增的整数
	[in]dwPTZCommand	控制命令类型
	[in]lParam1	缓存长度，单位：字节
	[in]lParam2	实际返回的数据长度，单位：字节
	[in]lParam3	查询状态等待时间，默认 1000ms，可根据需要自行设置
	[in]dwStop	停止标志。对云台八方向操作及镜头操作命令有效，进行其他操作时，本参数应填充 FALSE
	[in]param4	支持扩展控制命令参数
返回值	成功返回 TRUE 失败返回 FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

3.9 火情预警

3.9.1 订阅火点报警事件 CLIENT_RealLoadPictureEx

选项	说明	
描述	订阅火点报警事件	
函数	LLONG CLIENT_RealLoadPictureEx(LLONG lLoginID, int nChannelID, DWORD dwAlarmType, BOOL bNeedPicFile, fAnalyzerDataCallBack cbAnalyzerData, LDWORD dwUser, void* Reserved);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 的返回值
	[in] nChannelID	火点报警事件订阅通道
	[in] dwAlarmType	期望订阅的报警类型 火点报警事件：EVENT_IVS_FIREWARNING
	[in] bNeedPicFile	是否订阅图片文件
	[in] cbAnalyzerData	智能图片报警回调函数
	[in] dwUser	用户数据，SDK 通过智能图片报警回调函数 fAnalyzerDataCallBack 将该数据返回给用户，以便用户后续操作
	[out]Reserved	保留参数
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.9.2 停止订阅火点报警事件 CLIENT_StopLoadPic

选项	说明	
描述	停止订阅火点报警事件	
函数	BOOL CLIENT_StopLoadPic(LLONG lAnalyzerHandle);	
参数	[in] lAnalyzerHandle	火点报警事件订阅 ID

选项	说明
返回值	成功返回 TRUE，失败返回 FALSE
说明	无

3.9.3 设置火点信息回调函数接口 CLIENT_SetDVRMessCallBack

选项	说明	
描述	设置火点信息回调函数接口	
函数	void CLIENT_SetDVRMessCallBack(fMessCallBack cbMessage, LDWORD dwUser);	
参数	[in] cbMessage	报警回调函数
	[in] dwUser	用户数据
返回值	无	
说明	CLIENT_SetDVRMessCallBack 接口需在报警订阅之前调用，设置的回调函数里面接收到的事件不会有包含图片的事件	

3.9.4 订阅火点信息接口 CLIENT_StartListenEx

选项	说明	
描述	订阅火点信息接口	
函数	BOOL CLIENT_StartListenEx(LLONG lLoginID);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 的返回值
返回值	成功返回 TRUE，失败返回 FALSE	
说明	所有设备的报警事件都是通过 CLIENT_SetDVRMessCallBack 接口设置的回调函数反馈给用户的	

3.9.5 停止订阅火点信息接口 CLIENT_StopListen

选项	说明	
描述	停止订阅火点信息接口	
函数	BOOL CLIENT_StopListen(LLONG lLoginID);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 的返回值
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.10 冷热点获取 RadiometryGetCurrentHotColdSpotInfo

选项	说明
描述	获取热成像当前冷（最低的温度）、热（最高的温度）点信息

选项	说明	
函数	BOOL CALL_METHOD CLIENT_RadiometryGetCurrentHotColdSpotInfo(LLONG lLoginID, const NET_IN_RADIOMETRY_CURRENTHOTCOLDSPOT_INFO* pInParam, NET_OUT_RADIOMETRY_CURRENTHOTCOLDSPOT_INFO* pOutParam, int nWaitTime);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
	[in] pInParam	接口输入参数，内存资源由用户申请和释放
	[out] pOutParam	接口输出参数，内存资源由用户申请和释放
	[in]waittime	查询状态等待时间，默认 1000ms，可根据需要自行设置
返回值	成功返回 TRUE 失败返回 FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

3.11 随机区域温度信息 RadiometryGetRandomRegionTemper

选项	说明	
描述	获取测温区域的参数值	
函数	BOOL CALL_METHOD CLIENT_RadiometryGetRandomRegionTemper(LLONG lLoginID, const NET_IN_RADIOMETRY_RANDOM_REGION_TEMPER* pInParam, NET_OUT_RADIOMETRY_RANDOM_REGION_TEMPER* pOutParam, int nWaitTime);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
	[in] pInParam	接口输入参数，内存资源由用户申请和释放
	[out] pOutParam	接口输出参数，内存资源由用户申请和释放
	[in] nWaitTime	查询状态等待时间，默认 1000ms，可根据需要自行设置
返回值	成功返回 TRUE 失败返回 FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

3.12 测温设置

3.12.1 设备配置信息获取 CLIENT_GetNewDevConfig

选项	说明
描述	获取通道名称

选项	说明	
函数	BOOL CLIENT_GetNewDevConfig(LLONG lLoginID, char* szCommand, int nChannelID, char* szOutBuffer, DWORD dwOutBufferSize, lnt *error, int waittime=500);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
	[in] szCommand	命令参数，
	[in] nChannelID	设备通道号，从 0 开始
	[out] szOutBuffer	输出缓冲
	[out] dwOutBufferSize	输出缓冲大小
	[out] error	错误码 ● 0：成功 ● 1：失败 ● 2：数据不合法 ● 3：暂时无法设置 ● 4：没有权限
	[in] nWaitTime	等待超时时间，默认 500ms
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.12.2 解析查询到的配置信息 CLIENT_ParseData

选项	说明	
描述	解析查询到的配置信息	
函数	BOOL CLIENT_ParseData(char* szCommand, char* szInBuffer, LPVOID lpOutBuffer, DWORD dwOutBufferSize, void* pReserved);	
参数	[in] szCommand	命令参数
	[in] szInBuffer	输入缓存
	[out] lpOutBuffer	输出缓存
	[in] dwOutBufferSize	输出缓冲大小
	[in] pReserved	保留
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.12.3 组成要设置的配置信息 CLIENT_PacketData

选项	说明
描述	组成要设置的配置信息
函数	BOOL CLIENT_PacketData(char* szCommand,

选项	说明	
	LPVOID lpInBuffer, DWORD dwInBufferSize, char* szOutBuffer, DWORD dwOutBufferSize);	
参数	[in] szCommand	命令参数
	[in] lpInBuffer	输入缓存
	[in] dwInBufferSize	输入缓存大小
	[out]szOutBuffer	输出缓存
	[in] dwOutBufferSize	输出缓存大小
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.12.4 设备配置信息设置 CLIENT_SetNewDevConfig

选项	说明	
描述	设备配置信息设置	
函数	BOOL CLIENT_SetNewDevConfig(LLONG lLoginID, char* szCommand, int nChannelID, char* szInBuffer, DWORD dwInBufferSize, int *error, int *restart, int waittime=500);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
	[in] szCommand	命令参数，
	[in] nChannelID	设备通道号，从 0 开始
	[in] szInBuffer	输入缓存，用户存储组成的用于设置配置的 json 串信息
	[in] dwInBufferSize	缓存地址大小
	[out] error	错误码 ● 0：成功 ● 1：失败 ● 2：数据不合法 ● 3：暂时无法设置 ● 4：没有权限
	[in] restart	重启标志地址
	[in] waitTime	等待超时时间，默认 500ms
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.12.5 设置报警回调函数 CLIENT_SetDVRMessCallBack

选项	说明
描述	设置报警回调函数

选项	说明	
函数	void CLIENT_SetDVRMessCallBack(fMessCallBack cbMessage, LDWORD dwUser);	
参数	[in] cbMessage	● 消息回调函数，可以回调设备的状态，如报警状态 ● 当设置为 0 时表示禁止回调
	[in] dwUser	用户自定义数据
返回值	无	
说明	● 设置设备消息回调函数，用来得到设备当前状态信息，与调用顺序无关，SDK 默认不回调 ● 此回调函数 fMessCallBack 必须先调用报警消息订阅接口 CLIENT_StartListenEx 才生效	

3.12.6 订阅报警 CLIENT_StartListenEx

选项	说明	
描述	订阅报警	
函数	BOOL CLIENT_StartListenEx(LLONG lLoginID);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
返回值	成功返回 TRUE，失败返回 FALSE	
说明	订阅设备消息，得到的消息从 CLIENT_SetDVRMessCallBack 的设置值回调出来	

普通报警事件类型说明请参见表 3-5。

表3-6 普通事件类型说明

dwAlarmType 宏定义	宏定义值	含义	回调 pAlarmInfo 对应结构体
DH_ALARM_ALARM_EX	0x2101	外部报警	DWORD
DH_MOTION_ALARM_EX	0x2102	动态检测	DWORD
DH_ALARM_ALARM_EX2	0x2175	本地报警事件	ALARM_ALARM_INFO_EX2
DH_ALARM_TEMPERATURE	0x2135	温度过高报警	ALARM_TEMPERATURE_INFO
DH_ALARM_STORAGE_FAILURE_EX	0x2163	存储错误报警	ALARM_STORAGE_FAILURE_EX
DH_ALARM_STORAGE_NOT_EXIST	0x3167	存储组不存在事件	ALARM_STORAGE_NOT_EXIST_INFO
DH_ALARM_ALARM_EX2	0x2175	本地报警事件	ALARM_ALARM_INFO_EX2
DH_ALARM_HEATIMG_TEMPER	0x31aa	热成像测温点温度异常报警事件	ALARM_HEATIMG_TEMPER_INFO
DH_ALARM_FIREWARNING	0x31b5	热成像着火点事件	ALARM_FIREWARNING_INFO
DH_ALARM_SMOKE_DETECTION	0X31d5	烟雾报警事件	ALARM_SMOKE_DETECTION_INFO
DH_ALARM_BETWEENRULE_TEMP_DIFF	0x31d6	热成像规则间温差异常报警	ALARM_BETWEENRULE_DIFFTEMPER_INFO

dwAlarmType 宏定义	宏定义值	含义	回调 pAlarmInfo 对应结构体
DH_ALARM_HOTSPOT_WARNING	0X31d8	热成像热点异常报警	ALARM_HOTSPOT_WARNING_INFO
DH_ALARM_COLDSPOT_WARNING	0X31d9	热成像冷点异常报警	ALARM_COLDSPOT_WARNING_INFO
DH_ALARM_FIREWARNING_INFO	0X31da	热成像火情信息上报	ALARM_FIREWARNING_INFO_DETAIL
DH_ALARM_FACE_OVERHEATING	0x31db	热成像人体发烧预警（人脸超温报警）	ALARM_FACE_OVERHEATING_INFO

3.12.7 停止订阅报警 CLIENT_StopListen

选项	说明	
描述	停止订阅报警	
函数	BOOL CLIENT_StopListen(LLONG lLoginID);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

3.12.8 获取规则温度信息 CLIENT_RadiometryGetCurTemperAll

选项	说明	
描述	获取当前所有规则温度信息	
函数	BOOL CALL_METHOD CLIENT_RadiometryGetCurTemperAll(LLONG lLoginID, const NET_IN_RADIOMETRY_GET_CUR_TEMPER_ALL_INFO* pInParam, NET_OUT_RADIOMETRY_GET_CUR_TEMPER_ALL_INFO* pOutParam, int nWaitTime);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
	[in] pInParam	接口输入参数，内存资源由用户申请和释放
	[out] pOutParam	接口输出参数，内存资源由用户申请和释放
	[in] nWaitTime	查询状态等待时间，默认 1000ms，可根据需要自行设置
返回值	成功返回 TRUE 失败返回 FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

3.12.9 订阅测温温度数据 CLIENT_RadiometryAttachTemper

选项	说明
描述	订阅测温温度数据

选项	说明	
函数	LLONG CALL_METHOD CLIENT_RadiometryAttachTemper(LLONG lLoginID, const NET_IN_RADIOMETRY_ATTACH_TEMPER* pInParam, NET_OUT_RADIOMETRY_ATTACH_TEMPER* pOutParam, int nWaitTime);	
参数	[in] lLoginID	CLIENT_LoginWithHighLevelSecurity 返回值
	[in] pInParam	接口输入参数，内存资源由用户申请和释放
	[out] pOutParam	接口输出参数，内存资源由用户申请和释放
	[in] nWaitTime	查询状态等待时间，默认 1000ms，可根据需要自行设置
返回值	成功返回 TRUE 失败返回 FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

3.12.10 取消订阅测温温度数据 CLIENT_RadiometryAttachTemper

选项	说明	
描述	取消订阅测温温度数据	
函数	BOOL CALL_METHOD CLIENT_RadiometryDetachTemper(LLONG lAttachTemperHandle);	
参数	[in] lAttachTemperHandle	处理句柄
返回值	成功返回 TRUE 失败返回 FALSE	
说明	接口返回失败，请使用 CLIENT_GetLastError 获取错误码	

第 4 章 回调函数

4.1 断线回调函数 fDisConnect

选项	说明	
描述	断线回调函数	
函数	typedef void (CALLBACK *fDisConnect)(LLONG lLoginID, char *pchDVRIP, LONG nDVRPort, LDWORD dwUser);	
参数	[out] lLoginID	CLIENT_LoginEx2 的返回值
	[out] pchDVRIP	断线的设备 IP
	[out] nDVRPort	断线的设备端口
	[out] dwUser	回调函数的用户参数
返回值	无	
说明	无	

4.2 断线重连回调函数 fHaveReConnect

选项	说明	
描述	断线重连回调函数	
函数	typedef void (CALLBACK *fHaveReConnect)(LLONG lLoginID, char *pchDVRIP, LONG nDVRPort, LDWORD dwUser);	
参数	[out] lLoginID	CLIENT_LoginEx2 的返回值
	[out] pchDVRIP	断线后重连成功的设备 IP
	[out] nDVRPort	断线后重连成功的设备端口
	[out] dwUser	回调函数的用户参数
返回值	无	
说明	无	

4.3 实时预览数据回调函数 fRealDataCallBackEx

选项	说明
描述	实时预览数据回调函数

选项	说明	
函数	typedef void (CALLBACK *fRealDataCallBackEx)(LLONG lRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, LONG param, LDWORD dwUser);	
参数	[out] lRealHandle	CLIENT_RealPlayEx 的返回值
	[out] dwDataType	数据类型，0 表示原始数据，2 表示 YUV 数据
	[out] pBuffer	预览数据块地址
	[out] dwBufSize	预览数据块的长度，单位：字节
	[out] param	回调数据参数结构体，dwDataType 值不同类型不同 ● dwDataType 为 0 时，param 为空指针 ● dwDataType 为 2 时，param 为 tagCBYUVDataParam 结构体指针
	[out] dwUser	回调函数的用户参数
返回值	无	
说明	无	

4.4 普通报警回调函数 fMessCallBack

选项	说明	
描述	报警回调函数	
函数	BOOL (CALLBACK *fMessCallBack)(LONG lCommand, LLONG lLoginID, char *pBuf, DWORD dwBufLen, char *pchDVRIP, LONG nDVRPort, LDWORD dwUser);	
参数	[out] lCommand	报警类型，具体请参见表 4-1
	[out] lLoginID	登录接口返回值
	[out] pBuf	接收报警数据的缓存，根据调用的侦听接口和 lCommand 值不同，填充的数据不同
	[out] dwBufLen	pBuf 的长度，单位：字节
	[out] pchDVRIP	设备 ip
	[out] nDVRPort	端口
	[out] dwUser	用户自定义数据
返回值	TRUE 回调函数执行正确，FALSE 执行错误	
说明	一般在应用程序初始化时调用设置回调，在回调函数中根据不同的设备 ID 和命令值做出不同的处理	

表4-1 报警类型说明

报警类型	报警类型名称	pBuf
DH_MOTION_ALARM_EX	动态检测报警	数据字节数与设备视频通道个数相同，每个字节表示一个视频通道的动态检测报警状态，1 为有报警，0 为无报警

报警类型	报警类型名称	pBuf
DH_ALARM_STORAGE_FAILURE	硬盘故障报警	ALARM_STORAGE_FAILURE 数组
EVENT_ALARM_VIDEOLOSS	视频丢失报警	无
DH_ALARM_FRONTDISCONNECT	IPC 断网报警	ALARM_FRONTDISCONNET_INFO
DH_ALARM_ALARM_EX	外部报警	数据字节数与设备报警通道个数相同，每个字节表示一个报警通道的报警状态，1 为有报警，0 为无报警

4.5 智能事件回调 fAnalyzerDataCallBack

选项	说明	
描述	智能事件回调	
函数	typedef int (CALLBACK *fAnalyzerDataCallBack)(LLONG lAnalyzerHandle, DWORD dwAlarmType, void* pAlarmInfo, BYTE *pBuffer, DWORD dwBufSize, LDWORD dwUser, int nSequence, void *reserved);	
参数	[out]lAnalyzerHandle	CLIENT_RealLoadPictureEx 返回值
	[out]dwAlarmType	智能事件类型
	[out]pAlarmInfo	事件信息缓存
	[out]pBuffer	图片缓存
	[out]dwBufSize	图片缓存大小
	[out]dwUser	用户数据
	[out]reserved	保留
返回值	无	
说明	无	

4.6 温度分布数据回调函数 fRadiometryAttachCB

选项	说明	
描述	温度分布数据回调函数	
函数	typedef void (CALLBACK *fRadiometryAttachCB)(LLONG lAttachHandle, NET_RADIOMETRY_DATA* pBuf, int nBufLen, LDWORD dwUser);	
参数	[out] lAttachHandle	CLIENT_RadiometryAttach 的返回值
	[out] pBuf	数据块地址
	[out] nBufLen	数据块长度，单位：字节
	[out] dwUser	回调函数的用户参数
返回值	无	
说明	无	

附录1 法律声明

商标声明

- HDMI™ HIGH-DEFINITION MULTIMEDIA INTERFACE：本声明适用所有产品。如本产品使用 HDMI 技术，词语 HDMI、HDMI High-Definition Multimedia Interface（高清晰度多媒体接口）、HDMI 商业外观和 HDMI 徽标均为 HDMI Licensing Administrator, Inc.的商标或注册商标。本产品已经获得 HDMI Licensing Administrator, Inc.授权使用 HDMI 技术。
- VGA 是 IBM 公司的商标。
- Windows 标识和 Windows 是微软公司的商标或注册商标。
- 在本文档中可能提及的其他商标或公司的名称，由其各自所有者拥有。

责任声明

- 在适用法律允许的范围内，在任何情况下，本公司都不对因本文档中相关内容及描述的产品而产生任何特殊的、附随的、间接的、继发性的损害进行赔偿，也不对任何利润、数据、商誉、文档丢失或预期节约的损失进行赔偿。
- 本文档中描述的产品均“按照现状”提供，除非适用法律要求，本公司对文档中的所有内容不提供任何明示或暗示的保证，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证。

隐私保护提醒

您安装了我们的产品，您可能会采集人脸、指纹、车牌等个人信息。在使用产品过程中，您需要遵守所在地区或国家的隐私保护法律法规要求，保障他人的合法权益。如，提供清晰、可见的标牌，告知相关权利人视频监控区域的存在，并提供相应的联系方式。

关于本文档

- 本文档供多个型号产品使用，产品外观和功能请以实物为准。
- 如果不按照本文档中的指导进行操作而造成的任何损失由使用方自己承担。
- 本文档会实时根据相关地区的法律法规更新内容，具体请参见产品的纸质、电子光盘、二维码或官网，如果纸质与电子档内容不一致，请以电子档为准。
- 本公司保留随时修改本文档中任何信息的权利，修改的内容将会在本文档的新版本中加入，恕不另行通知。
- 本文档可能包含技术上不准确的地方、或与产品功能及操作不相符的地方、或印刷错误，以公司最终解释为准。
- 如果获取到的 PDF 文档无法打开，请使用最新版本或最主流的阅读工具。

附录2 安全建议

账户管理

1. **使用复杂口令**
   建议口令设置如下：
   - 长度不小于 8 个字符。
   - 至少包含两种字符类型，字符类型包括大小写字母、数字和符号。
   - 不包含用户名或用户名的倒序。
   - 不要使用连续字符，如 123、abc 等。
   - 不要使用重叠字符，如 111、aaa 等。
2. **定期修改口令**
   建议您定期修改设备口令，以降低被猜测或破解的风险。
3. **合理分配账户及权限**
   根据业务和管理需要，合理新增账户，并为其分配最小权限集合。
4. **开启账户锁定**
   出厂默认开启账户锁定功能，建议您保持开启状态，以保护账户安全。在攻击者多次口令尝试失败后，其对应账户及源 IP 将会被锁定。
5. **及时设置、更新口令重置信息**
   设备支持口令重置功能，为了降低该功能被攻击者利用的风险，如有信息变更，请及时修改。设置密保问题时，建议不要使用容易猜测的答案。

服务配置

1. **使能 HTTPS**
   建议您开启 HTTPS，通过安全的通道访问 WEB 服务。
2. **音视频加密传输**
   如果您的音视频数据内容比较重要或敏感，建议启用加密传输功能，以降低音视频数据传输过程中被窃听的风险。
3. **关闭非必需服务，使用安全的模式**
   如果没有需要，建议您关闭 SSH、SNMP、SMTP、UPnP、AP 热点等功能，以降低攻击面。
   如果有需要，强烈建议您使用安全的模式，包括但不限于：
   - SNMP：选择 SNMP V3，并设置复杂的加密口令和鉴权口令。
   - SMTP：选择 TLS 方式接入邮箱服务器。
   - FTP：选择 SFTP，并设置复杂口令。
   - AP 热点：选择 WPA2-PSK 加密模式，并设置复杂口令。
4. **更改 HTTP 及其它服务默认端口**
   建议您将 HTTP 及其他服务默认端口更改为 1024～65535 间的任意端口，以减小攻击者猜测服务的风险。

网络配置

1. **启用白名单**
   建议您开启白名单功能，开启后仅允许白名单列表中的 IP 访问设备。因此，请务必将您的电脑 IP 地址，以及配套的设备 IP 地址加入白名单列表中。

2. **MAC 地址绑定**
   建议您在设备端将其网关的 IP 与 MAC 地址进行绑定，以降低 ARP 欺骗风险。
3. **安全网络环境的搭建**
   为了更好地保障设备的安全性，降低网络安全风险，建议您：
   - 关闭路由器端口映射功能，避免外部网络直接访问路由器内网设备服务。
   - 根据实际网络需要，对网络进行划区隔离：若两个子网间没有通信需求，建议使用 VLAN、网闸等方式对其进行网络分割，达到网络隔离效果。
   - 建立 802.1x 接入认证体系，以降低非法终端接入专网的风险。

安全审计

1. **查看在线用户**
   建议您不定期查看在线用户，识别是否有非法用户登录。
2. **查看设备日志**
   通过查看日志，可以获知尝试登录设备的 IP 信息，以及已登录用户的关键操作信息。
3. **配置网络日志**
   由于设备存储容量限制，日志存储能力有限，若您需要长期保存日志，建议您启用网络日志功能，确保关键日志同步至网络日志服务器，便于问题回溯。

软件安全

1. **固件及时更新**
   遵循行业的标准作业规范，设备固件需要及时更新至最新版本，以保证设备具有最新的功能和安全性。若设备接入公网时，建议开启在线升级自动检测功能，便于及时获知厂商发布的固件更新信息。
2. **客户端及时更新**
   建议您下载和使用最新版本客户端软件。

物理防护

建议您对设备（尤其是存储类设备）进行物理防护，比如将设备放置在专用机房或机柜，并做好门禁权限和钥匙管理，防止未经授权的人员进行破坏硬件、外接设备（例如 U 盘、串口）等物理攻击。