NetSDK 编程指导手册

（智能 AI 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 IVSS、NVR、IPC、ITC、客流量统计设备、闸机等设备的智能业务涉及的 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、产品经理和项目经理。

符号约定

在本文档中可能出现下列标志，代表的含义如下。

标识	说明
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	更新附录 1 智能事件列表	2025.02
V1.0.6	新增附录 3 智能事件。	2023.12
V1.0.5	全文优化语言。	2023.02

版本号	修订内容	发布日期
V1.0.4	● 增加部分通用行为事件。 ● 增加部分智能交通事件。 ● 新增物品监控事件。 ● 新增城管业务事件。 ● 新增车位检测事件。 ● 新增人群分布图事件。 ● 新增车辆密度图事件。 ● 新增热度图事件。 ● 新增立体行为分析事件。	2021.08
V1.0.3	● 删除 avnetsdk 依赖库信息。 ● 新增 StreamConvertor 依赖库。 ● 删除 fisheye 鱼眼矫正库。	2021.05
V1.0.2	修改录像或图片查询、人脸库信息查询的示例代码。	2020.08
V1.0.1	修改登录设备和搜索设备接口函数。	2020.03
V1.0.0	首次发布。	2018.10

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	解释
IVSS	Intelligent Video Surveillance System，区别传统 NVR 只进行存储的业务形态，加入智能化分析，形成一体化管理的系统。
目标检测	通过对视频进行智能分析，检测出其中的目标及目标的特征信息（年龄、性别，表情等）。
历史库	用于存放设备抓拍到的人脸图的库。
人脸库	通过预先导入一些人脸图片到 IVSS、NVR、前端 IPC 等设备中，实时检测人脸是否在其中。
以通道为对象进行布控	对某一通道进行一个或者多个人脸库的布控与撤控，应用场景：该通道检测到的人脸与布控库进行比对，并给出结果。属于人脸库布控的一种方式。
以库为对象进行布控	将某个人脸库布控到一个或者多个通道上，应用场景：该人脸库布控的通道检测到的人脸与该人脸库进行比对，对比完成后，给出结果。属于人脸库布控的一种方式。
以图搜图	外部导入一张图片和相似度值，IVSS、NVR 等设备通过这张图检索历史库或人脸库是否已经存在匹配的人脸，并返回在相似度之上的图片结果。
ITC	Intelligent Traffic Camera，智能交通摄像机。具有抓拍车辆图片并自动分析交通事件的功能。
绊线入侵检测	自动检测穿越警戒线的行为。
区域入侵检测	自动检测目标入侵警戒区的行为，包括“穿越区域”和“在区域内”。
客流量	相机标定区域内的人数信息。
人数统计	实时统计相机标定区域进入或者离开的人数。
区域内人数统计	实时统计相机标定区域内的人数。

目录

前言 I

名词解释 III

第 1 章 内容简介 1

1.1 概述 1

1.2 适用性 2

1.3 应用场景 2

1.3.1 客流量统计 2

1.3.2 智能交通 3

1.3.3 通用行为 4

1.3.4 闸机 5

第 2 章 通用功能 7

2.1 SDK 初始化 7

2.1.1 简介 7

2.1.2 接口总览 7

2.1.3 流程说明 7

2.1.4 示例代码 8

2.2 设备登录 9

2.2.1 简介 9

2.2.2 接口总览 10

2.2.3 流程说明 10

2.2.4 示例代码 12

2.3 实时预览 13

2.3.1 简介 13

2.3.2 接口总览 13

2.3.3 流程说明 13

2.3.4 示例代码 17
2.4 智能事件订阅 18
2.4.1 简介 18
2.4.2 接口总览 18
2.4.3 流程说明 19
2.4.4 示例代码 20
2.5 查询/回放/下载录像和图片 21
2.5.1 简介 21
2.5.2 接口总览 21
2.5.3 流程说明 21
2.5.4 示例代码 24
第 3 章 目标检测 31
3.1 人脸事件订阅 31
3.2 人脸库的增删改查 31
3.2.1 简介 31
3.2.2 接口总览 31
3.2.3 流程说明 31
3.2.4 示例代码 33
3.3 人脸的增删改查 36
3.3.1 简介 36
3.3.2 接口总览 36
3.3.3 流程说明 36
3.3.4 示例代码 38
3.4 以通道或者库为对象布控 42
3.4.1 简介 42
3.4.2 接口总览 42
3.4.3 流程说明 42
3.4.4 示例代码 44

3.5 以图搜图 47
3.5.1 简介 47
3.5.2 接口总览 47
3.5.3 流程说明 47
3.5.4 示例代码 49
3.6 人脸相关录像图片查询与下载 51
第 4 章 人体检测 53
4.1 人体事件订阅 53
4.2 人体图片搜索 53
4.2.1 简介 53
4.2.2 接口总览 53
4.2.3 流程说明 53
4.2.4 示例代码 54
第 5 章 客流量统计 56
5.1 客流量事件订阅 56
5.1.1 简介 56
5.1.2 接口总览 56
5.1.3 流程说明 56
5.1.4 示例代码 57
5.2 客流量事件报警 58
5.3 客流量历史数据查询 58
5.3.1 简介 58
5.3.2 接口总览 59
5.3.3 流程说明 59
5.3.4 示例代码 60
第 6 章 通用行为事件 62
6.1 通用行为事件订阅 62
6.2 通用行为事件录像查询和下载 62

第 7 章 智能交通 64
7.1 智能交通事件订阅 64
7.2 车流量历史数据查询 65
7.2.1 简介 65
7.2.2 接口总览 65
7.2.3 流程说明 65
7.2.4 示例代码 67
7.3 车辆黑白名单增删改查 69
7.3.1 简介 69
7.3.2 接口总览 69
7.3.3 流程说明 70
7.3.4 示例代码 71
7.4 车辆相关图片的查询与下载 74
7.4.1 简介 74
7.4.2 接口总览 74
7.4.3 流程说明 74
7.4.4 示例代码 76
第 8 章 闸机 79
8.1 门禁事件订阅 79
8.2 门禁卡信息管理 79
8.2.1 简介 79
8.2.2 接口总览 79
8.2.3 流程说明 79
8.2.4 示例代码 81
8.3 人脸管理 87
8.3.1 简介 87
8.3.2 接口总览 87
8.3.3 流程说明 87

8.3.4 示例代码 89
8.4 门禁进出记录查询 92
8.4.1 简介 92
8.4.2 接口总览 93
8.4.3 流程说明 93
8.4.4 示例代码 95
第 9 章 物品监控 96
9.1 物品监控事件订阅 96
9.2 物品监控事件录像查询和下载 96
第 10 章 城管业务 97
10.1 城管业务事件订阅 97
10.2 城管业务事件录像查询和下载 97
第 11 章 车位检测任务 99
11.1 车位检测事件订阅 99
11.2 车位检测事件录像查询和下载 99
11.3 订阅指定的车位图片信息 100
11.3.1 简介 100
11.3.2 接口总览 100
11.3.3 流程说明 100
11.3.4 示例代码 101
第 12 章 人群分布图 103
12.1 人群分布事件订阅 103
12.2 人群分布事件录像查询和下载 103
第 13 章 车辆密度图 104
13.1 车辆密度事件订阅 104
13.2 车辆密度事件录像查询和下载 104
第 14 章 热度图 105
14.1 订阅热度图数据 105

14.1.1 简介 105
14.1.2 接口总览 105
14.1.3 流程说明 105
14.1.4 示例代码 106
14.2 订阅灰度图数据 107
14.2.1 简介 107
14.2.2 接口总览 107
14.2.3 流程说明 108
14.2.4 示例代码 108
第 15 章 立体行为分析 110
15.1 立体行为分析事件订阅 110
15.2 立体行为分析事件录像查询和下载 110
15.3 订阅立体行为-视频统计摘要信息 111
15.3.1 简介 111
15.3.2 接口总览 111
15.3.3 流程说明 112
15.3.4 示例代码 113
第 16 章 安全帽检测 114
16.1 安全帽检测事件订阅 114
第 17 章 接口函数 114
17.1 SDK 初始化 114
17.1.1 SDK 初始化 CLIENT_Init 114
17.1.2 SDK 清理 CLIENT_Cleanup 115
17.1.3 设置断线重连回调函数 CLIENT_SetAutoReconnect 115
17.1.4 设置网络参数 CLIENT_SetNetworkParam 115
17.2 设备登录 116
17.2.1 高安全级别登录 CLIENT_LoginWithHighLevelSecurity 116
17.2.2 用户登出设备 CLIENT_Logout 118

17.3 实时预览 118
17.3.1 打开预览 CLIENT_RealPlayEx 118
17.3.2 关闭预览 CLIENT_StopRealPlayEx 120
17.3.3 保存预览数据 CLIENT_SaveRealData 120
17.3.4 停止保存预览数据 CLIENT_StopSaveRealData 121
17.3.5 设置预览数据回调 CLIENT_SetRealDataCallBackEx 121
17.4 智能事件订阅 122
17.4.1 智能事件订阅 CLIENT_RealLoadPictureEx 122
17.4.2 智能事件的取消订阅 CLIENT_StopLoadPic 124
17.5 智能事相关录像图片查询与下载 124
17.5.1 按条件查询图片或录像 CLIENT_FindFileEx 124
17.5.2 获取查询到文件个数 CLIENT_GetTotalFileCount 125
17.5.3 查找文件 CLIENT_FindNextFileEx 125
17.5.4 结束文件查找 CLIENT_FindCloseEx 126
17.5.5 开始录像回放 CLIENT_PlayBackByTimeEx2 127
17.5.6 结束录像回放 CLIENT_StopPlayBack 127
17.5.7 开始录像下载 CLIENT_DownloadByTimeEx 128
17.5.8 停止录像下载 CLIENT_StopDownload 129
17.5.9 下载图片 CLIENT_DownloadRemoteFile 129
17.6 人脸事件订阅 130
17.7 人脸库的增删改查 130
17.7.1 人脸库的增删改操作 CLIENT_OperateFaceRecognitionGroup 130
17.7.2 人脸库信息查询 CLIENT_FindGroupInfo 131
17.8 人脸的增删改查 132
17.8.1 人脸的增删改操作 CLIENT_OperateFaceRecognitionDB 132
17.8.2 设置人脸的查询条件 CLIENT_StartFindFaceRecognition 133
17.8.3 查询人脸 CLIENT_DoFindFaceRecognition 133
17.8.4 结束人脸查找 CLIENT_StopFindFaceRecognition 134

17.8.5 以库为对象进行布控 CLIENT_FaceRecognitionPutDisposition 134
17.8.6 以库为对象进行撤控 CLIENT_FaceRecognitionDelDisposition 135
17.8.7 以通道为对象进行布控 CLIENT_SetGroupInfoForChannel 136
17.8.8 订阅人脸查询进度 CLIENT_AttachFaceFindState 137
17.8.9 取消订阅人脸查询进度 CLIENT_DetachFaceFindState 138
17.9 人体检测 138
17.9.1 人体图片下载 CLIENT_DownloadRemoteFile 138
17.10 客流量统计 139
17.10.1 客流量事件订阅 CLIENT_AttachVideoStatSummary 139
17.10.2 取消订阅客流量事件 CLIENT_DetachVideoStatSummary 139
17.10.3 开始查询客流历史数据（设置查询条件）CLIENT_StartFindNumberStat 140
17.10.4 查询客流历史数据 CLIENT_DoFindNumberStat 141
17.10.5 结束查询历史数据 CLIENT_StopFindNumberStat 141
17.11 智能交通 142
17.11.1 开始查找数据（设置查询条件）CLIENT_FindRecord 142
17.11.2 查询数据总数 CLIENT_QueryRecordCount 142
17.11.3 查询指定条数数据 CLIENT_FindNextRecord 143
17.11.4 结束车流量查询 CLIENT_FindRecordClose 143
17.11.5 黑白名单的增删改 CLIENT_OperateTrafficList 144
17.11.6 批量下载文件 CLIENT_DownLoadMultiFile 145
17.11.7 停止批量下载文件 CLIENT_StopLoadMultiFile 145
17.12 门禁 146
17.12.1 开始查找数据（设置查询条件）CLIENT_FindRecord 146
17.12.2 查询指定条数数据 CLIENT_FindNextRecord 146
17.12.3 结束查询 CLIENT_FindRecordClose 147
17.12.4 人员/门禁记录信息操作 CLIENT_ControlDevice 147
17.12.5 人脸图片的操作 CLIENT_FaceInfoOpreate 148
17.13 车位检测任务 149

17.13.1 订阅指定的车位图片信息 149
17.13.2 取消订阅指定的车位图片信息 150
17.14 热度图 150
17.14.1 订阅热度图数据 150
17.14.2 取消订阅热度图数据 151
17.14.3 订阅灰度图数据 151
17.14.4 取消订阅灰度图数据 152
17.15 立体行为分析 152
17.15.1 订阅立体行为-视频统计摘要信息 152
17.15.2 取消订阅立体行为-视频统计摘要信息 153
第 18 章 回调函数定义 154
18.1 断线回调函数 fDisConnect 154
18.2 断线重连回调函数 fHaveReConnect 154
18.3 实时预览数据回调函数 fRealDataCallBackEx 155
18.4 智能事件回调函数 fAnalyzerDataCallBack 156
18.5 回放及按文件下载进度回调函数 fDownLoadPosCallBack 157
18.6 回放及下载数据回调函数 fDataCallBack 158
18.7 人脸查询进度回调函数 fFaceFindState 159
18.8 客流量事件订阅回调 fVideoStatSumCallBack 159
18.9 批量下载文件进度回调函数 fMultiFileDownLoadPosCB 160
18.10 指定车位图片信息事件订阅回调函数 161
18.11 热度图原始数据事件订阅回调函数 162
18.12 热度图灰度数据事件订阅回调函数 163
18.13 立体行为视频统计摘要信息事件订阅回调函数 163
附录 1 智能事件 165
附录 2 法律声明 211
附录 3 网络安全建议 212

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。

主要功能包括通用功能、目标检测、人体检测、客流量统计、通用行为事件、智能交通和闸机。

根据环境不同，开发包包含的文件会不同，具体如下所示。

- Windows 开发包所包含的文件，请参见表 1-1。

表1-1 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	头文件
	dhconfigsdk.lib	Lib 文件
	dhconfigsdk.dll	库文件
播放（编码解码）辅助库	dhplay.dll	播放库
dhnetsdk 辅助库	IvsDrawer.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 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

库类型	库文件名称	库文件说明
	dhconfigsdk.h	头文件
	libdhconfigsdk.so	库文件
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、配置以及获取和处理码流数据等。
- 配置库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 支持的系统如下：
  - ◇ Windows

    Windows 10/Windows 8.1/Windows 7/vista/2000 以及 Windows Server 2008/2003。
  - ◇ Linux

    Red Hat/SUSE 等通用 Linux 系统。

1.3 应用场景

1.3.1 客流量统计

客流产品在实际场景的应用，如图 1-1 所示。

图1-1 客流场景



1.3.2 智能交通

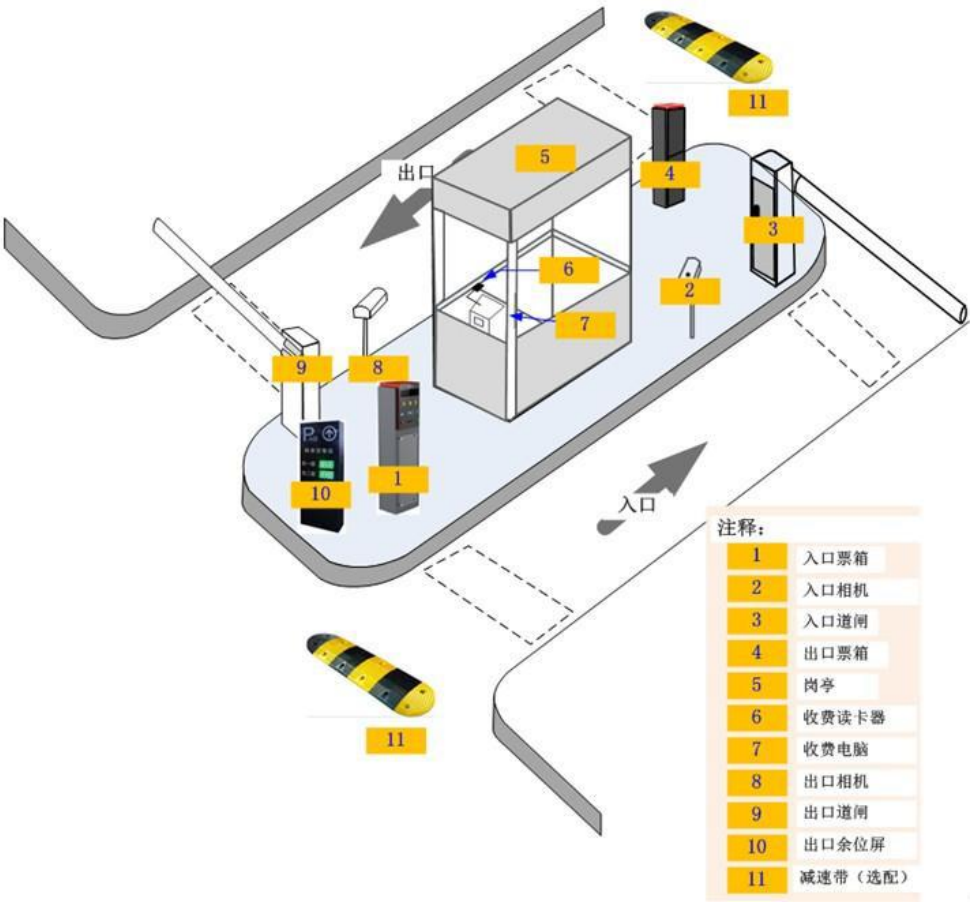
ITC 在交通路口的应用，用于抓拍交通违章行为及车辆流量统计，如图 1-2 所示。

图1-2 ITC 在交通路口的应用



ITC 在停车场出入口的应用，用于控制车辆进出停车场及监控车位是否有空余，如图 1-3 所示。

图1-3 ITC 在停车场出入口的应用



1.3.3 通用行为

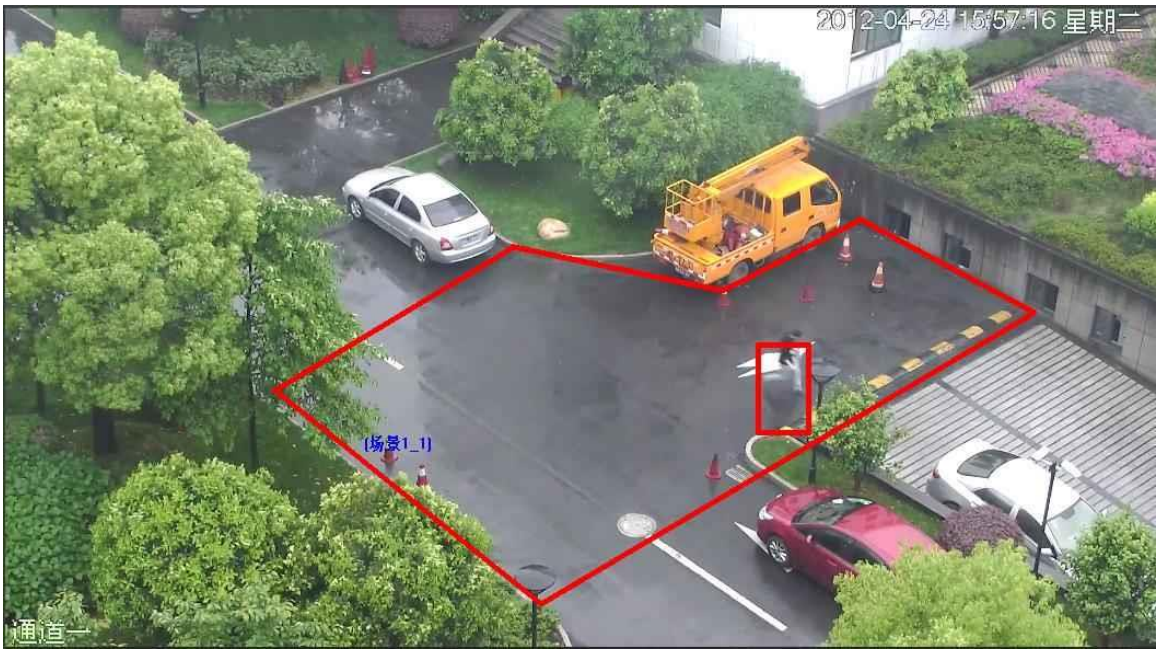
指人或者车辆穿越规则线（绊线入侵）或者入侵警戒区（区域入侵）产生相应的报警事件，同时可以区分出目标物体（人或者车辆）。

通用行为应用场景如图 1-4、1-8 所示。

图1-4 通用行为场景-绊线入侵



图1-5 通用行为场景-区域入侵



1.3.4 闸机

门禁闸机主要应用园区、景区、学校、小区、办公楼等。将采集的人脸照片和人员信息上传平台后，再由平台将数据下发到闸机系统。

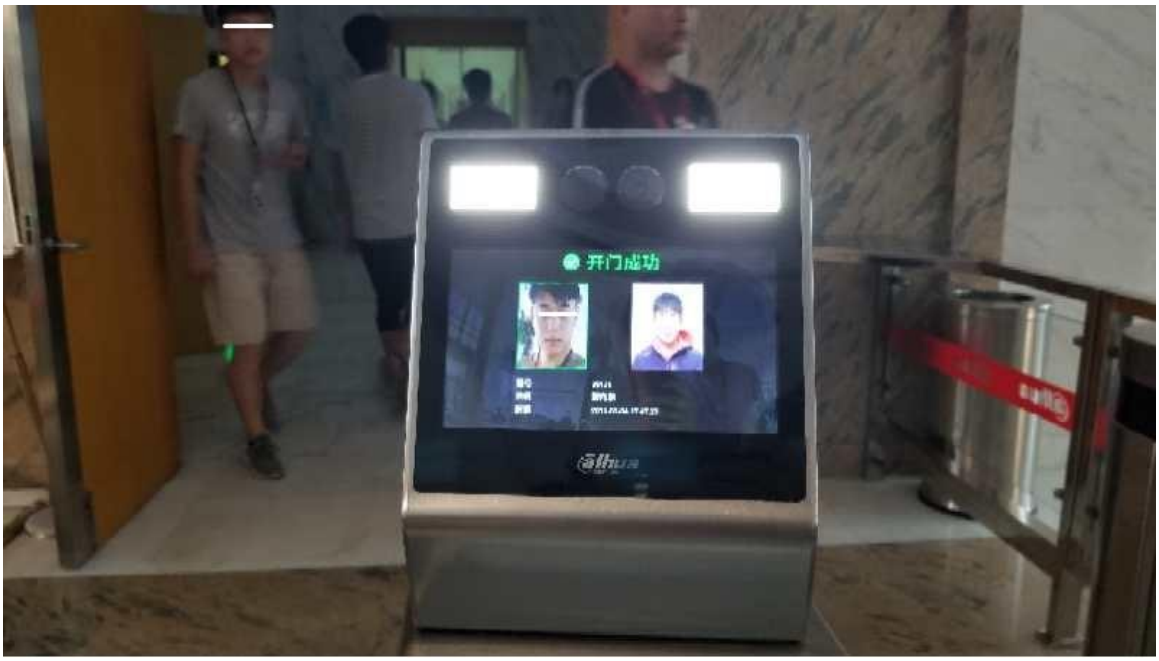
门禁闸机外观如图 1-6 所示。

图1-6 摆闸外观



闸机可以通过人脸或者刷卡的方式进行开门，人脸开门如图 1-7 所示。

图1-7 人脸开门



第 2 章 通用功能

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

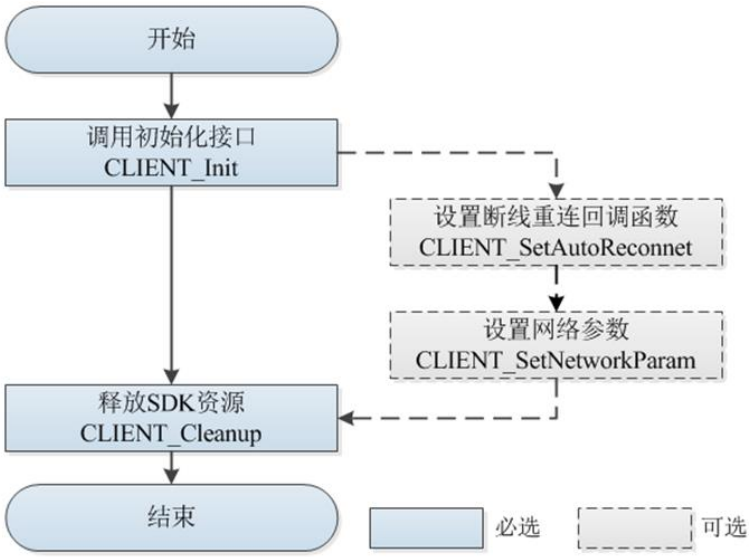
表2-1 SDK 初始化接口信息

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、报警和智能图片订阅业务，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

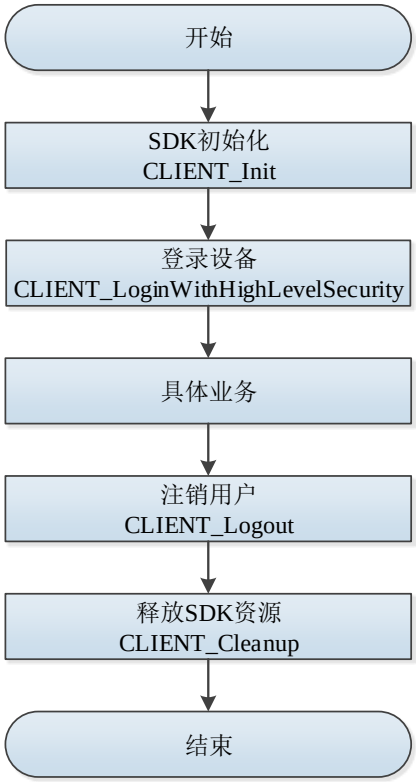
表2-2 设备登录接口信息

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。
- 句柄重复：登录句柄有可能与存在过的句柄相同，属于正常现象。例如登录设备 A 获得 loginIDA，将 loginIDA 注销，再次进行登录操作，可能又获取到 LoginIDA。但是在句柄的整个生命周期内，不会出现重复的句柄。
- 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 sokcet 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
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

error 的错误码	对应的含义
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，如果网络状况较差，容易出现错误码 3 时，可以使用以下代码增大超时时间：

```
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_SetRealDataCallBackEx	设置实时预览数据回调函数扩展接口

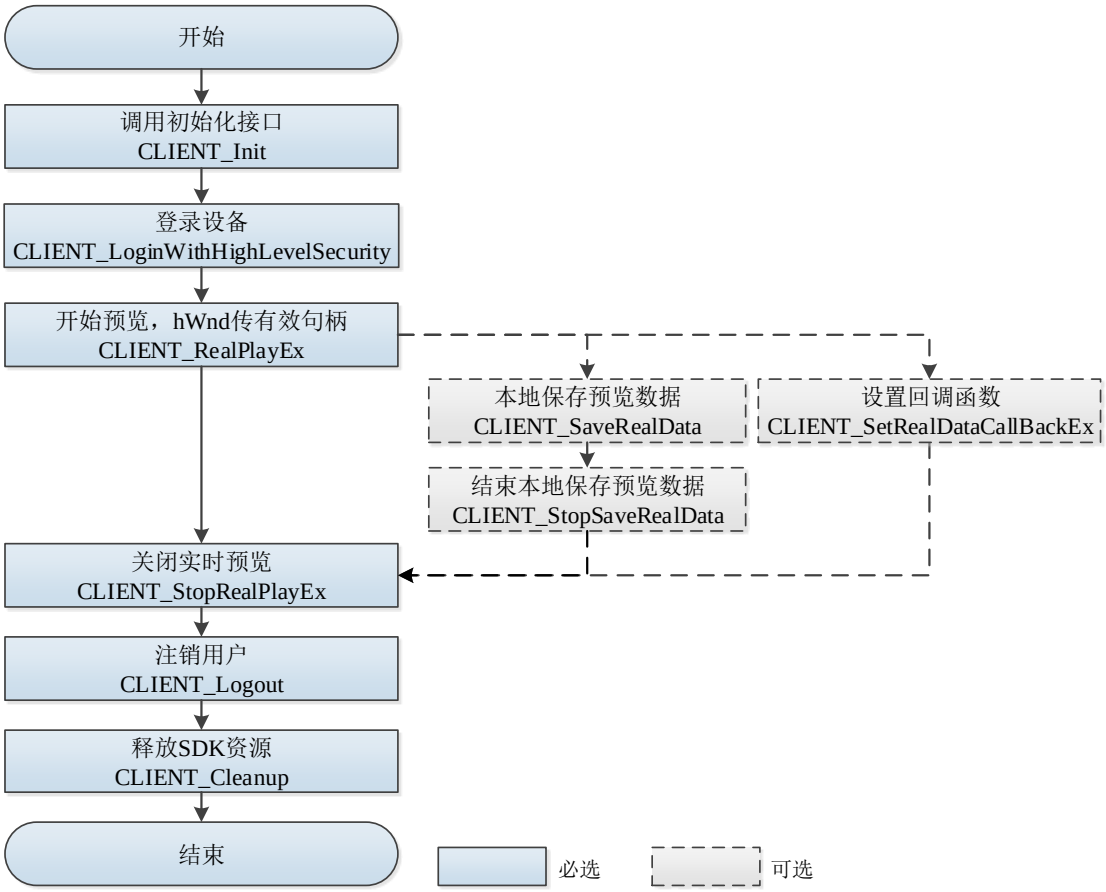
2.3.3 流程说明

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 集成播放库播放

SDK 内部调用辅助库里的 PlaySDK 库实现实时播放。SDK 集成播放库解码播放流程，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_SetRealDataCallBackEx，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** 的值修改得大一些。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; // unit ms

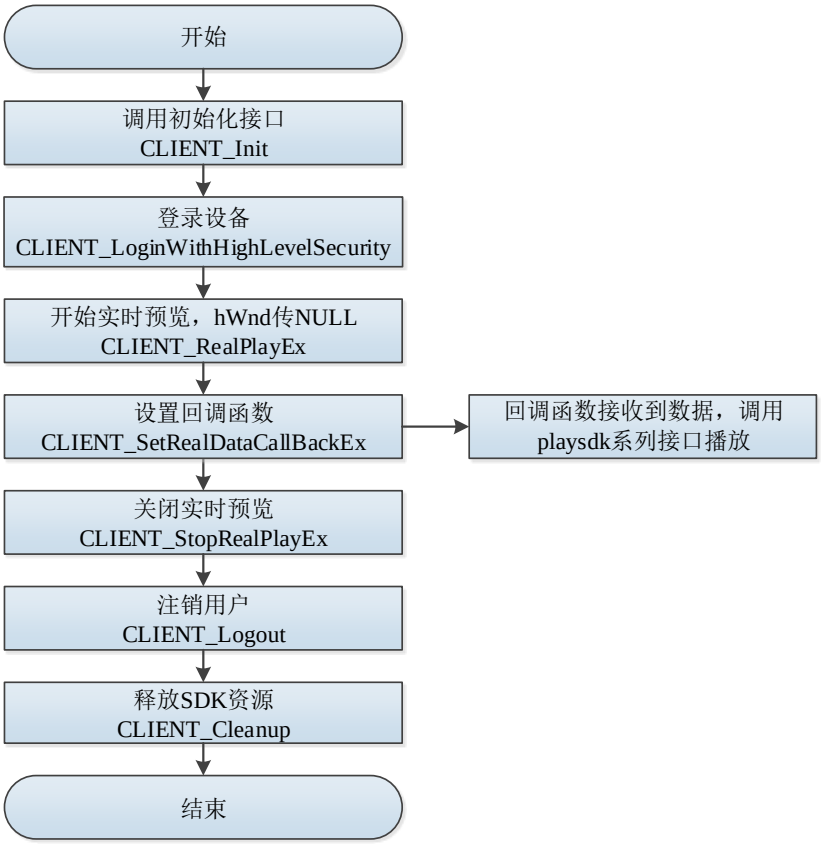
CLIENT_SetNetworkParam (&stuNetParam);
```

- 重复打开失败：部分设备不支持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 调用私有播放库”。

2.3.3.2 调用私有播放库播放

SDK 回调实时预览码流给用户，用户调用 PlaySDK 进行解码播放。用户调用私有播放库解码播放流程如图 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_SetRealDataCallBackEx 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（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 集成播放库播放

```
// 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 调用私有播放库播放

```
// 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 = 0x00000001;
    CLIENT_SetRealDataCallBackEx(lRealHandle, &RealDataCallBackEx, NULL, dwFlag);
}
// 关闭预览
if (0 != lRealHandle)
{
CLIENT_StopRealPlayEx(lRealHandle);
}

void CALLBACK RealDataCallBackEx(LLONG lRealHandle, DWORD dwDataType, BYTE *pBuffer,
DWORD dwBufSize, LONG 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 智能事件订阅

2.4.1 简介

智能事件订阅，即前端设备智能算法或者后端智能算法对码流实时进行分析，当检测到预先设定好的智能事件时，就将该事件和事件信息上报给用户。在本文档中的智能事件包括，通用行为分析智能事件（绊线入侵、区域入侵等智能事件）、目标检测、人体检测、智能交通的智能事件（卡口、超速、低速、交通拥堵等智能事件）。

2.4.2 接口总览

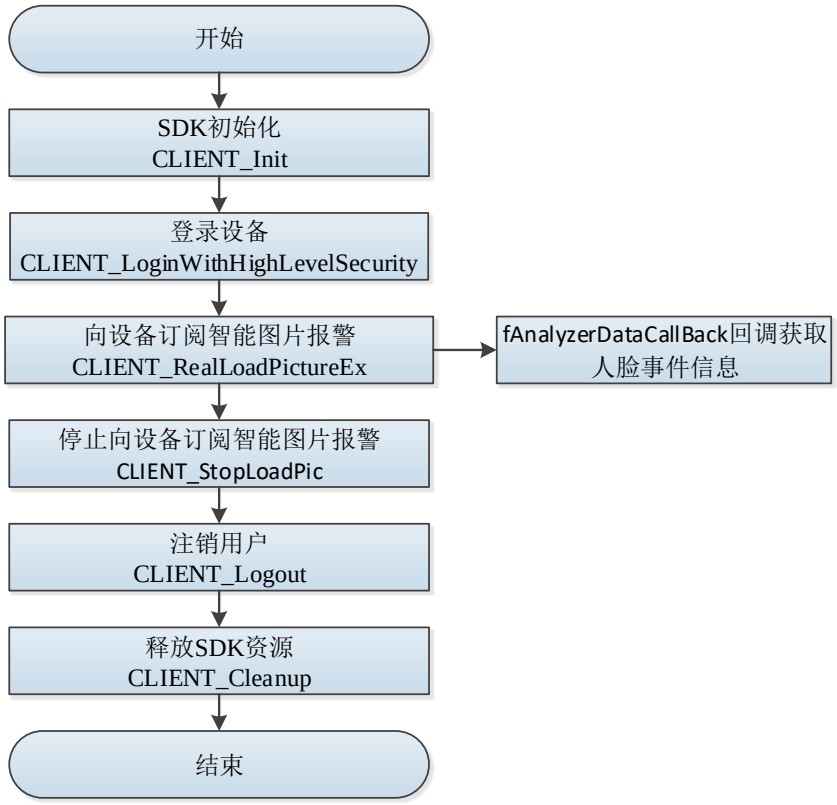
表2-5 智能事件订阅接口说明

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

2.4.3 流程说明

人脸事件上报流程，如图 2-5 所示。

图2-5 人脸事件上报流程



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。

2.4.4 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_FACERECOGNITION:

……

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);
```

2.5 查询/回放/下载录像和图片

2.5.1 简介

设备的智能算法在分析实时码流时，如果检测为某个智能事件，就会触发对该智能事件的录像和抓拍，并将其保存。用户可以查询设备中保存下来的智能事件的录像和图片，并对查询到的结果进行下载或者回放操作。

2.5.2 接口总览

表2-6 查询/回放/下载录像和图片接口说明

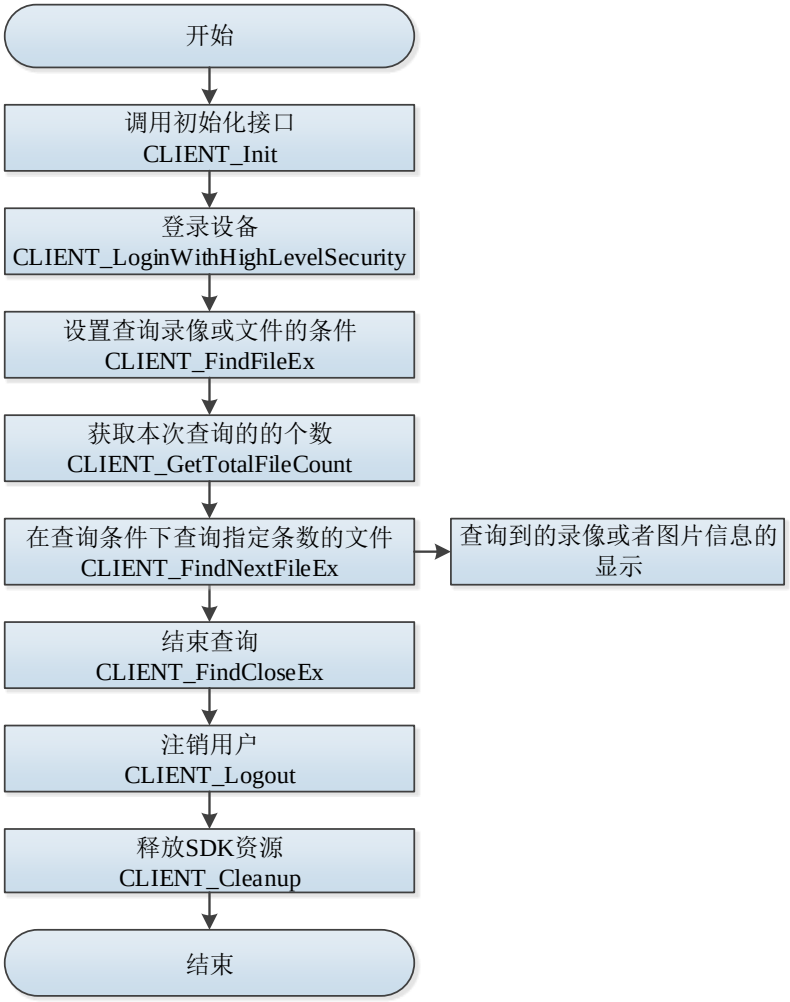
接口	说明
CLIENT_FindFileEx	按条件查询录像或者图片，设置查询条件
CLIENT_GetTotalFileCount	获取本次查询的录像或者图片的个数
CLIENT_FindNextFileEx	查询指定的条数的录像或者图片
CLIENT_FindCloseEx	结束查询
CLIENT_PlayBackByTimeEx2	按时间开始录像的回放
CLIENT_StopPlayBack	停止录像回放
CLIENT_DownloadByTimeEx	录像下载
CLIENT_StopDownload	停止录像下载
CLIENT_DownloadRemoteFile	下载图片

2.5.3 流程说明

2.5.3.1 录像或图片的查询流程

录像或图片的查询流程，如图 2-6 所示。

图2-6 录像或图片的查询流程



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　调用 CLIENT_FindFileEx 设置查询条件，设置成功后返回查询句柄。根据 emType 的不同取值，判断查找的类型。

步骤4　调用 CLIENT_GetTotalFileCount 获取查询到的录像或者文件总数。

步骤5　调用 CLIENT_FindNextFileEx 查询指定条数的录像或者图片，并将查询到的信息保存，用于录像或者图片的回放下载操作。

步骤6　调用 CLIENT_FindCloseEx 结束查询。

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。

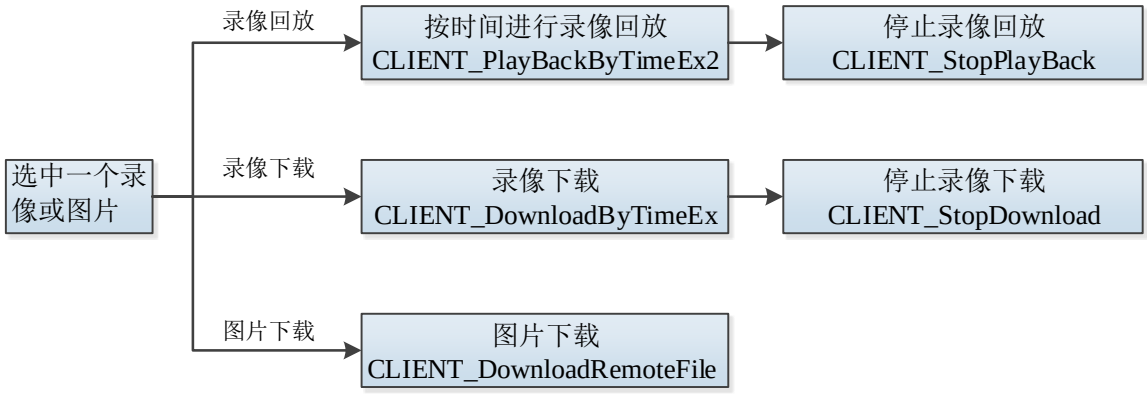
注意事项

- 接口 CLIENT_FindFileEx 中参数 pQueryCondition 是用户申请和释放的，具体类型由 emType 的枚举类型来定义。
- CLIENT_FindFileEx 如果查询成功，则会返回查询句柄，CLIENT_FindNextFileEx 将查询句柄作为参数查询具体的录像或者图片，且必须调用 CLIENT_FindCloseEx 将查询句柄关闭。
- CLIENT_FindNextFileEx 可设置查询条数，如果设置条数大于 1 条则参数 pMediaFileInfo 必须为一个数据指针，数据大小大于等于设置的条数。

2.5.3.2 录像的回放和下载、图片的下载流程

录像的回放和下载、图片的下载流程，如图 2-7 所示。

图2-7 录像的回放和下载、图片的下载流程



流程说明

选择一条接口 CLIENT_FindNextFileEx 查询到的结果信息，然后选择下载还是回放。

- 录像回放

  如果是录像文件，使用录像查询结果中的开始时间和结束时间，调用接口 CLIENT_PlayBackByTimeEx2 进行回放。回放过程中或者回放结束后，调用接口 CLIENT_StopPlayBack 进行录像回放的停止。

- 录像下载

  如果是录像文件，使用录像查询结果中的开始时间和结束时间，调用接口 CLIENT_DownloadByTimeEx 进行录像的下载。下载结束后，调用 CLIENT_StopDownload 接口进行录像下载的停止。

- 图片下载

  如果选择的是图片文件，使用图片查询结果中的文件名和图片类型，调用接口

CLIENT_DownloadRemoteFile 下载图片。

注意事项

录像的回放与下载，以及图片的下载都需要依赖录像和图片的查询结果，并以查询到信息作为回放和下载的条件。

2.5.4 示例代码

2.5.4.1 录像或图片的查询

```
// 查询条件
MEDIAFILE_FACE_DETECTION_PARAM param;
memset(&param, 0, sizeof(param));
param.dwSize = sizeof(param);
param.stuDetail.dwSize = sizeof(MEDIAFILE_FACE_DETECTION_DETAIL_PARAM);
param.nChannelID = -1;
param.stuStartTime = startTime;
param.stuEndTime = endTime
param.emPicType = NET_FACEPIC_TYPE_SMALL;  // 人脸小图
param.bDetailEnable = FALSE;
param.emSex = EM_DEV_EVENT_FACEDETECT_SEX_TYPE_MAN;
param.bAgeEnable = FALSE;
param.nEmotionValidNum = 0;
param.emGlasses = EM_FACEDETECT_WITH_GLASSES;


// 查询目标检测的小图
LLONG lFindFileHandle = CLIENT_FindFileEx(g_lLoginHandle, DH_FILE_QUERY_FACE_DETECTION,
&param, NULL,5000);
if (lFindFileHandle == 0)
{
printf("CLIENT_FindFileEx: failed! Error code: %x.\n", CLIENT_GetLastError());
```

```
return;
}

// 获取查询到的人脸个数
BOOL nRet = CLIENT_GetTotalFileCount(lFindFileHandle,&nCount,NULL);
if (!nRet)
{
printf("CLIENT_GetTotalFileCount: failed! Error code: %x.\n", CLIENT_GetLastError());
return;
}

// 查询个数
int nMaxConut = 10;
MEDIAFILE_FACE_DETECTION_INFO* pMediaFileInfo = NEW
MEDIAFILE_FACE_DETECTION_INFO[nMaxConut];
memset(pMediaFileInfo, 0, sizeof(MEDIAFILE_FACE_DETECTION_INFO) * nMaxConut);
for (int i = 0; i < nMaxConut; i++)
{
    pMediaFileInfo[i].dwSize = sizeof(MEDIAFILE_FACE_DETECTION_INFO);
}

// 开始查询
int nRet = CLIENT_FindNextFileEx(lFindFileHandle, nMaxConut, (void*)pMediaFileInfo,
nMaxConut * sizeof(MEDIAFILE_FACE_DETECTION_INFO), NULL,3000);
if (nRet < 0)
{
printf("CLIENT_FindNextFileEx: failed! Error code: %x.\n", CLIENT_GetLastError());
return;
}
```

```
// 关闭查询
CLIENT_FindCloseEx(lFindFileHandle);
```

2.5.4.2 录像回放

```
// 设置回放时的码流类型，此处设置成主码流
int nStreamType = 0; // 0-主辅码流，1-主码流，2-辅码流
CLIENT_SetDeviceMode(lLoginHandle, DH_RECORD_STREAM_TYPE, &nStreamType);
// 设置回放时的录像文件类型，此处设置成所有录像
NET_RECORD_TYPE emFileType = NET_RECORD_TYPE_ALL; // 所有录像
CLIENT_SetDeviceMode(lLoginHandle, DH_RECORD_TYPE, &emFileType);
// 开启录像回放
int nChannelID = 0; // 通道号
NET_IN_PLAY_BACK_BY_TIME_INFO stIn = {0};
NET_OUT_PLAY_BACK_BY_TIME_INFO stOut = {0};
memcpy(&stIn.stStartTime, &stuStartTime, sizeof(stuStartTime));
memcpy(&stIn.stStopTime, &stuStopTime, sizeof(stuStopTime));
stIn.hWnd = hWnd;
stIn.fDownLoadDataCallBack = DataCallBack;
stIn.dwDataUser = NULL;
stIn.cbDownLoadPos = NULL;
stIn.dwPosUser = NULL;
stIn.nPlayDirection = emDirection;
stIn.nWaittime = 10000;
LLONG lPlayHandle = CLIENT_PlayBackByTimeEx2(lLoginHandle, nChannelID, &stIn, &stOut);
if (0 == lPlayHandle)
{
printf("CLIENT_PlayBackByTimeEx2: failed! Error code: %x.\n", CLIENT_GetLastError());
}
```

```
if (FALSE == CLIENT_StopPlayBack(lPlayHandle))
{
printf("CLIENT_StopPlayBack Failed, lRealHandle[%x]!Last Error[%x]\n" , lPlayHandle,
CLIENT_GetLastError());
}
```

2.5.4.3 录像下载

```
//回放进度函数
void CALLBACK TimeDownLoadPosCallBack(LLONG lPlayHandle, DWORD dwTotalSize, DWORD
dwDownLoadSize, int index, NET_RECORDFILE_INFO recordfileinfo, LDWORD dwUser);

// 回放/下载数据回调函数
int CALLBACK DataCallBack(LLONG lRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD
dwBufSize, LDWORD dwUser);

int main()
{
// 设置查询时的录像码流类型，此处设置码流类型为主辅码流
int nStreamType = 0; // 0-主辅码流，1-主码流，2-辅码流
CLIENT_SetDeviceMode(lLoginHandle, DH_RECORD_STREAM_TYPE, &nStreamType);

// 设置下载开始和结束时间
int nChannelID = 0; // 通道号

NET_TIME stuStartTime = {0};
stuStartTime.dwYear = 2018;
stuStartTime.dwMonth = 9;
stuStartTime.dwDay = 17;
```

```
NET_TIME stuStopTime = {0};
stuStopTime.dwYear = 2018;
stuStopTime.dwMonth = 9;
stuStopTime.dwDay = 18;

// 开启录像下载
// 函数形参 sSavedFileName 和 fDownLoadDataCallBack 需至少有一个为有效值，否则入参有误
lDownloadHandle = CLIENT_DownloadByTimeEx(lLoginHandle, nChannelID,
EM_RECORD_TYPE_ALL, &stuStartTime, &stuStopTime, "test.dav", TimeDownLoadPosCallBack,
NULL, DataCallBack, NULL);
if (lDownloadHandle == 0)
{
printf("CLIENT_DownloadByTimeEx: failed! Error code: %x.\n", CLIENT_GetLastError());
}

// 关闭下载，可在下载结束后调用，也可在下载中调用
if (0 != lDownloadHandle)
{
if (!CLIENT_StopDownload(lDownloadHandle))
{
printf("CLIENT_StopDownload Failed, lDownloadHandle[%x]!Last Error[%x]\n" ,
lDownloadHandle, CLIENT_GetLastError());
}
}
}

void CALLBACK TimeDownLoadPosCallBack(LLONG lPlayHandle, DWORD dwTotalSize, DWORD
```

```
dwDownLoadSize, int index, NET_RECORDFILE_INFO recordfileinfo, LDWORD dwUser)
{
// 用户对进度回调做处理
}

int CALLBACK DataCallBack(LLONG lRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD
dwBufSize, LDWORD dwUser)
{
switch(dwDataType)
{
case 0:
//Original data
// 用户在此处保存码流数据，离开回调函数后再进行解码或转发等一系列处理
break;
case 1://Standard video data
break;
case 2:   //yuv data
break;
case 3://pcm audio data
break;
default:
break;
}
return 0;
}
```

2.5.4.4 图片下载

```
DH_IN_DOWNLOAD_REMOTE_FILE stuRemoteFileParm;
memset(&stuRemoteFileParm, 0, sizeof(DH_IN_DOWNLOAD_REMOTE_FILE));
```

```
stuRemoteFileParm.dwSize = sizeof(DH_IN_DOWNLOAD_REMOTE_FILE);

stuRemoteFileParm.pszFileName = pInfo->stObjectPic.szFilePath ;

stuRemoteFileParm.pszFileDst = szFileName;


DH_OUT_DOWNLOAD_REMOTE_FILE *fileinfo = NEW DH_OUT_DOWNLOAD_REMOTE_FILE;

fileinfo->dwSize = sizeof(DH_OUT_DOWNLOAD_REMOTE_FILE);


if (!CLIENT_DownloadRemoteFile(g_lLoginHandle, &stuRemoteFileParm, fileinfo))

{

printf("CLIENT_DownloadRemoteFile Failed,Last Error[%x]\n" , CLIENT_GetLastError());

}
```

第 3 章 目标检测

3.1 人脸事件订阅

具体的订阅请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出目标检测事件，即 EVENT_IVS_FACEDETECT 目标检测事件和 EVENT_IVS_FACERECOGNITION 目标识别事件。

3.2 人脸库的增删改查

3.2.1 简介

人脸库即人脸图片和人员信息的一个集合，提供了人脸库的添加、删除、修改和查找功能。

3.2.2 接口总览

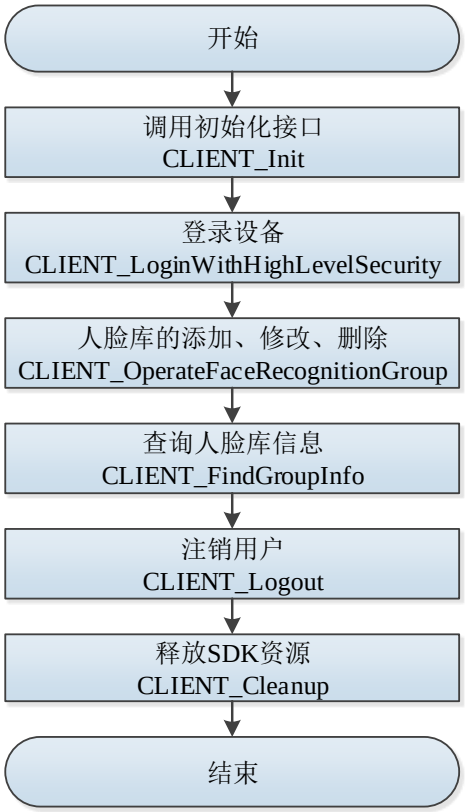
表3-1 人脸库的增删改查接口说明

接口	说明
CLIENT_OperateFaceRecognitionGroup	人脸库的添加、修改和删除操作
CLIENT_FindGroupInfo	人脸库信息的查询

3.2.3 流程说明

人脸库操作流程，如图 3-1 所示。

图3-1 人脸库操作流程



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_OperateFaceRecognitionGroup 根据相应的枚举类型进行人脸库的添加、修改、删除操作。

步骤4　调用 CLIENT_FindGroupInfo 获取人脸库信息。

步骤5　业务使用完后，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

- 人脸库添加：添加操作时，操作类型 emOperateType 对应的值取 NET_FACERECONGNITION_GROUP_ADD，所对应的结构体为 NET_ADD_FACERECONGNITION_GROUP_INFO。
- 人脸库修改：修改人脸库操作类型 emOperateType 对应值为 NET_FACERECONGNITION_GROUP_MODIFY，对应结构体为 NET_MODIFY_FACERECONGNITION_GROUP_INFO；修改人脸库需要指定 GroupID 和人脸库类型 emFaceDBType，所指定的 GroupID 需要设备上已存在的。

- 人脸库删除：删除人脸库操作类型 emOperateType 对应值为 NET_FACERECONGNITION_GROUP_DELETE，对应结构体为 NET_DELETE_FACERECONGNITION_GROUP_INFO；删除人脸库时若指定了 GroupID，则删除对应 GroupID 的人脸库，若未指定 GroupID，则删除全部的人脸库。

3.2.4 示例代码

3.2.4.1 人脸库信息的查询

```
// 设置人脸库查询条件
NET_IN_FIND_GROUP_INFO stuInParam = {sizeof(stuInParam)};
NET_OUT_FIND_GROUP_INFO stuOutParam = {sizeof(stuOutParam)};
stuOutParam.nMaxGroupNum = 100;
NET_FACERECONGNITION_GROUP_INFO *pGroupInfo = NULL;
stuOutParam.pGroupInfos = new NET_FACERECONGNITION_GROUP_INFO[100];
memset(stuOutParam.pGroupInfos, 0,sizeof(NET_FACERECONGNITION_GROUP_INFO)*100);
for (int i = 0; i < 100; i++)
{
   stuOutParam.pGroupInfos[i].dwSize = sizeof(FACERECONGNITION_GROUP_INFO);
}

// 人脸库查询
BOOL bRet = CLIENT_FindGroupInfo(lLoginHandle, &stuInParam, &stuOutParam, 5000);
if(FALSE == bRet)
{
printf("CLIENT_FindGroupInfo: failed! Error code %x.\n", CLIENT_GetLastError());
return -1;
}
delete[] pGroupInfo;
```

3.2.4.2 人脸库的增删改

```
enum EM_OPERATION_TYPE
{
FACEDB_DELETE,   // 删除
FACEDB_ADD,      // 添加
FACEDB_MODIFY    // 修改
};
// 设置需要删除的人脸库 ID
NET_FACERECONGNITION_GROUP_INFO *pstGroupInfo = m_pstSelectGroup;
NET_IN_OPERATE_FACERECONGNITION_GROUP stuInParam = {sizeof(stuInParam)};
NET_OUT_OPERATE_FACERECONGNITION_GROUP stuOutParam = {sizeof(stuOutParam)};
NET_ADD_FACERECONGNITION_GROUP_INFO stuAddGroupInfo = {sizeof(stuAddGroupInfo)};
NET_MODIFY_FACERECONGNITION_GROUP_INFO stuEditGroupInfo = {sizeof(stuEditGroupInfo)};

EM_OPERATION_TYPE emType = mType;
switch(emType)
{
// 删除人脸库
case FACEDB_DELETE:
{
stuInParam.emOperateType = NET_FACERECONGNITION_GROUP_DELETE;
NET_DELETE_FACERECONGNITION_GROUP_INFO stuDeleteInfo;
memset(&stuDeleteInfo, 0, sizeof(stuDeleteInfo));
stuDeleteInfo.dwSize = {sizeof(stuDeleteInfo)};
strncpy(stuDeleteInfo.szGroupId, pstGroupInfo->szGroupId, sizeof(stuDeleteInfo.szGroupId)-1);
stuInParam.pOPerateInfo = &stuDeleteInfo;
break;
}
```

```
// 添加人脸库
case FACEDB_ADD:
{
stuInParam.emOperateType = NET_FACERECONGNITION_GROUP_ADD;
stuAddGroupInfo.stuGroupInfo.dwSize = sizeof(stuAddGroupInfo.stuGroupInfo);
stuAddGroupInfo.stuGroupInfo.emFaceDBType = NET_FACE_DB_TYPE_BLACKLIST;

strncpy(stuAddGroupInfo.stuGroupInfo.szGroupName,pcGroupName,sizeof(stuAddGroupInfo.stuGroupInfo.szGroupName)-1);
stuInParam.pOPerateInfo = &stuAddGroupInfo;
break;
}
// 修改人脸库
case FACEDB_MODIFY:
{
stuInParam.emOperateType = NET_FACERECONGNITION_GROUP_MODIFY;
stuEditGroupInfo.stuGroupInfo.dwSize = sizeof(stuEditGroupInfo.stuGroupInfo);
stuEditGroupInfo.stuGroupInfo.emFaceDBType = NET_FACE_DB_TYPE_BLACKLIST;
strncpy(stuEditGroupInfo.stuGroupInfo.szGroupName, pcGroupName,
sizeof(stuEditGroupInfo.stuGroupInfo.szGroupName)-1);
strncpy(stuEditGroupInfo.stuGroupInfo.szGroupId, m_stuGroupInfo.szGroupId,
sizeof(stuEditGroupInfo.stuGroupInfo.szGroupId)-1);
stuInParam.pOPerateInfo = &stuEditGroupInfo;
break;
}
default:
break;
}
BOOL bRet = CLIENT_OperateFaceRecognitionGroup(m_lLoginID, &stuInParam, &stuOutParam,
5000);
```

```
if(FALSE == bRet)
{
printf("CLIENT_ OperateFaceRecognitionGroup: failed! Error code %x.\n", CLIENT_GetLastError());
return -1;
}
```

3.3 人脸的增删改查

3.3.1 简介

人脸库中存放着相应的人员即人脸信息，通过调用该功能接口可以完成人脸信息的添加、修改、查找、删除操作。

3.3.2 接口总览

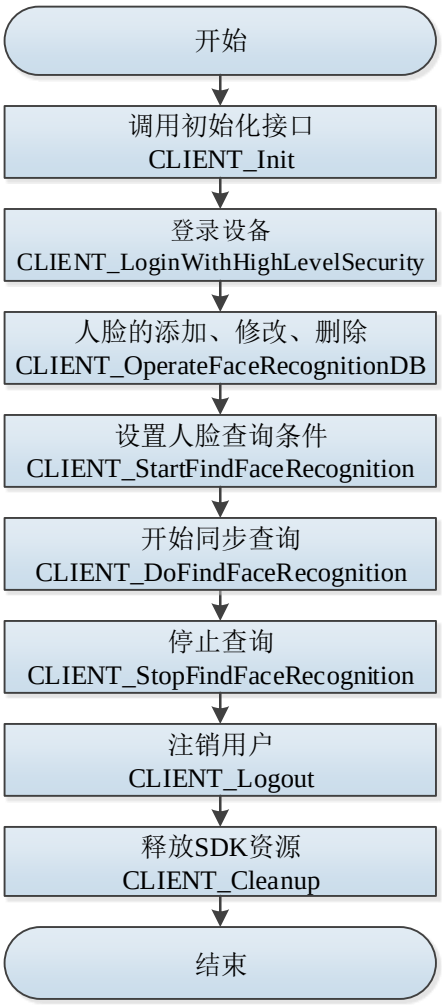
表3-2 人脸库的增删改查接口说明

接口	说明
CLIENT_OperateFaceRecognitionDB	人脸的添加、修改和删除操作
CLIENT_StartFindFaceRecognition	设置人脸的查询条件
CLIENT_DoFindFaceRecognition	查找指定条数的人脸数据
CLIENT_StopFindFaceRecognition	结束查询

3.3.3 流程说明

人脸的增删改查流程，如图 3-2 所示。

图3-2 人脸的增删改查流程



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_OperateFaceRecognitionDB 根据相应的枚举类型进行人脸库的添加、修改、删除操作。

步骤4　调用 CLIENT_StartFindFaceRecognition 设置人脸查询条件。

步骤5　调用 CLIENT_DoFindFaceRecognition 接口获取查询结果。

步骤6　调用 CLIENT_StopFindFaceRecognition 接口结束查询。

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

- 人脸库人脸添加：添加操作时，操作类型 emOperateType 对应的值取 NET_FACERECONGNITIONDB_ADD，入参结构体中 pBuffer 保存需要添加的人脸图片，用户自己申请和释放内存。GroupID 必填，其他按需填写。若添加人员成功，设备会返回 UID，即人员唯一标识符，用于表示唯一人员，见 pstOutParam -> szUID。
- 人脸库人脸修改：修改人脸操作类型 emOperateType 对应值为 NET_FACERECONGNITIONDB_MODIFY，入参结构体中 pBuffer 保存需要置换的人脸图片（设备上原有的样本图片删除，替换成现在新下发的图片），用户自己申请和释放内存；stPersonInfo 需要修改的人员信息，GroupID（可通过“查询当前人脸库信息”得到）和 szUID（人员唯一标识符，可通过“查找人员信息”）必须填写，其他按需填写。
- 人脸库删除：删除人脸库操作类型 emOperateType 对应值为 NET_FACERECONGNITIONDB_DELETE；stPersonInfo 需要删除的人员信息，GroupID（可通过“查询当前人脸库信息”得到）和 szUID（人员唯一标识符，可通过“查找人员信息”）。

3.3.4 示例代码

3.3.4.1 人脸的增删改

```
// 人员的增加与修改
NET_IN_OPERATE_FACERECONGNITIONDB stuInParam = { sizeof(stuInParam) };
NET_OUT_OPERATE_FACERECONGNITIONDB stuOutParam = { sizeof(stuOutParam) };
// 添加人员信息
{
stuInParam.emOperateType = NET_FACERECONGNITIONDB_ADD;
}

// 修改人员信息
{
//stuInParam.emOperateType = NET_FACERECONGNITIONDB_MODIFY;
//strncpy(stuInParam.stPersonInfoEx.szUID, strUID, sizeof(stuInParam.stPersonInfoEx.szUID) - 1);
}
```

```
stuInParam.bUsePersonInfoEx = TRUE;
stuInParam.stPersonInfoEx.bySex = 1;   // 男
stuInParam.stPersonInfoEx.byIDType = 1;   // 身份证
stuInParam.stPersonInfoEx.wYear = time.GetYear();
stuInParam.stPersonInfoEx.byMonth = time.GetMonth();
stuInParam.stPersonInfoEx.byDay = time.GetDay();
strncpy(stuInParam.stPersonInfoEx.szPersonName,
pstrName,sizeof(stuInParam.stPersonInfoEx.szPersonName) - 1);
strncpy(stuInParam.stPersonInfoEx.szID, pstrCardID, sizeof(stuInParam.stPersonInfoEx.szID) - 1);
strncpy(stuInParam.stPersonInfoEx.szGroupName, m_szGroupName,
sizeof(stuInParam.stPersonInfoEx.szGroupName) - 1);
strncpy(stuInParam.stPersonInfoEx.szGroupID, m_szGroupId,
sizeof(stuInParam.stPersonInfoEx.szGroupID) - 1);
stuInParam.nBufferLen = nPictureBufferLen;
stuInParam.pBuffer = pPictureBuffer;
stuInParam.stPersonInfoEx.wFacePicNum = 1;
stuInParam.stPersonInfoEx.szFacePicInfo[0].dwOffSet = 0;
stuInParam.stPersonInfoEx.szFacePicInfo[0].dwFileLenth = nLength;

bRet = CLIENT_OperateFaceRecognitionDB(m_lLoginID, &stuInParam, &stuOutParam, 5000);
if (FACE_PERSON_ADD == m_nOpreateType)
{
printf("CLIENT_OperateFaceRecognitionDB failed! Error code %x.\n", CLIENT_GetLastError());
return -1;
}



// 删除人员信息
```

```
NET_IN_OPERATE_FACERECONGNITIONDB stuInParam = { sizeof(stuInParam) };

NET_OUT_OPERATE_FACERECONGNITIONDB stuOutParam = { sizeof(stuOutParam) };

// 只需要 szGroupID 和 szUID 信息，其他不需要配置

stuInParam.emOperateType = NET_FACERECONGNITIONDB_DELETE;

stuInParam.bUsePersonInfoEx = TRUE;

strncpy(stuInParam.stPersonInfoEx.szUID,
m_pstPersonSelectInfo->stuCandidate.stPersonInfo.szUID,sizeof(stuInParam.stPersonInfoEx.szUID
) - 1);

strncpy(stuInParam.stPersonInfoEx.szGroupID, m_szGroupId,
sizeof(stuInParam.stPersonInfoEx.szGroupID) - 1);


BOOL bRet = CLIENT_OperateFaceRecognitionDB(lLoginHandle, &stuInParam, &stuOutParam,
5000);

if (!bRet)

{

printf("CLIENT_OperateFaceRecognitionDB failed! Error code %x.\n",CLIENT_GetLastError());

return -1;

}
```

3.3.4.2 人脸的查询

```
// 设置人脸查询条件

NET_IN_STARTFIND_FACERECONGNITION stuInParam = {sizeof(stuInParam)};

NET_OUT_STARTFIND_FACERECONGNITION stuOutParam = {sizeof(stuOutParam)};


stuInParam.stMatchOptions.dwSize = sizeof(stuInParam.stMatchOptions);

stuInParam.stFilterInfo.dwSize = sizeof(stuInParam.stFilterInfo);

stuInParam.bPersonExEnable = TRUE;

stuInParam.stFilterInfo.nRangeNum = 1;

stuInParam.stFilterInfo.szRange[0] = (BYTE)NET_FACE_DB_TYPE_BLACKLIST;

strncpy(stuInParam.stPersonInfoEx.szPersonName, m_PersonName,
```

```
sizeof(stuInParam.stPersonInfoEx.szPersonName)-1);
stuInParam.stPersonInfoEx.bySex = 0;
stuInParam.stFilterInfo.stBirthdayRangeStart = BirthdayRangeStart;
stuInParam.stFilterInfo.stBirthdayRangeEnd = BirthdayRangeEnd;
strncpy(stuInParam.stPersonInfoEx.szID, pcCard, sizeof(stuInParam.stPersonInfoEx.szID)-1);
strncpy(stuInParam.stFilterInfo.szGroupId[0], m_szGroupId, sizeof(m_szGroupId)-1);
stuInParam.stFilterInfo.nGroupIdNum = 1;
strncpy(stuInParam.stPersonInfoEx.szGroupID, m_szGroupId,
sizeof(stuInParam.stPersonInfoEx.szGroupID)-1);

BOOL bRet = CLIENT_StartFindFaceRecognition(m_lLoginID, &stuInParam, &stuOutParam,
DEFAULT_WAIT_TIME);
if (!bRet)
{
printf("CLIENT_StartFindFaceRecognition failed! Error code %x.\n", CLIENT_GetLastError());
return;
}

// 开始进行查询
NET_IN_DOFIND_FACERECONGNITION stuInDoFind = {sizeof(stuInDoFind)};
NET_OUT_DOFIND_FACERECONGNITION stuOutDoFind = {sizeof(stuOutDoFind)};
stuOutDoFind.bUseCandidatesEx = TRUE;

stuInDoFind.lFindHandle = m_lFindPersonHandle;
stuInDoFind.emDataType = EM_NEEDED_PIC_TYPE_HTTP_URL;
stuInDoFind.nCount = 10;
stuInDoFind.nBeginNum = m_nCurPos;
bRet = CLIENT_DoFindFaceRecognition(&stuInDoFind, &stuOutDoFind, WAIT_TIMEOUT);
if (!bRet)
```

```
{
    printf("CLIENT_DoFindFaceRecognition failed! Error code %x.\n", CLIENT_GetLastError());
    return;
}
```

3.4 以通道或者库为对象布控

3.4.1 简介

以通道为对象进行布控，即一个通道可布控一个或者多个人脸库。以库为对象进行布控，即一个人脸库可布控一个或者多个通道。两种方式都为人脸库的布控。

3.4.2 接口总览

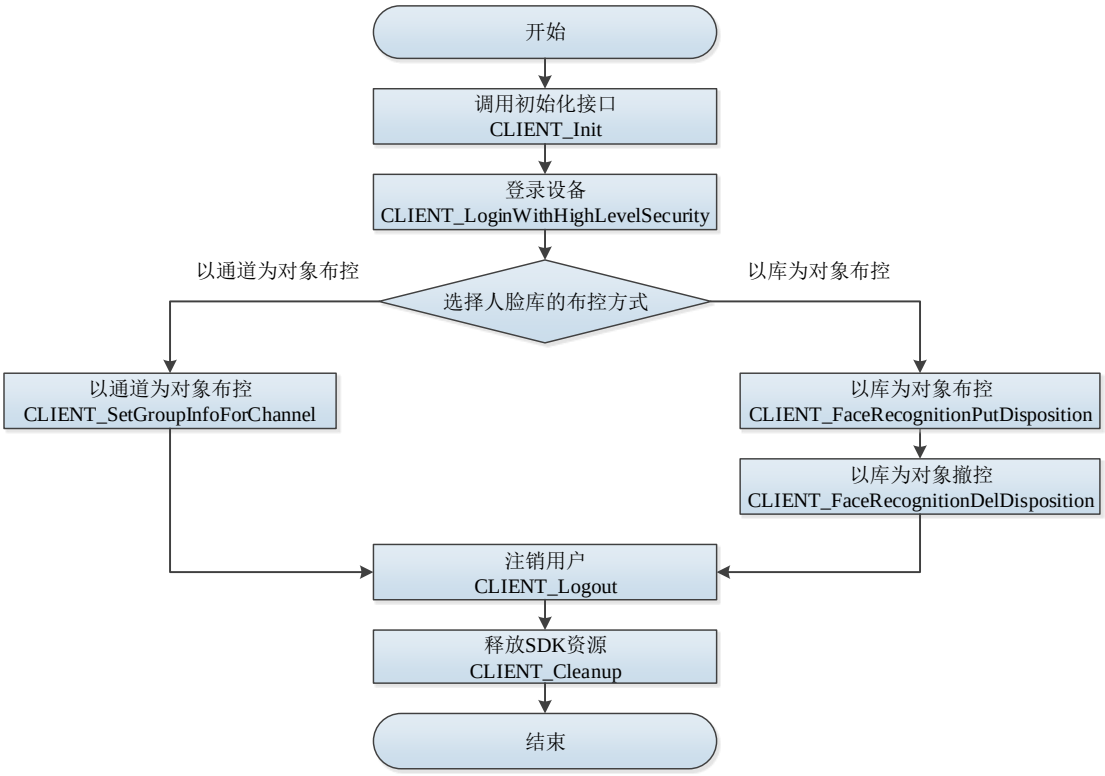
表3-3 以通道或者库为对象布控接口说明

接口	说明
CLIENT_FaceRecognitionPutDisposition	以库为对象进行布控
CLIENT_FaceRecognitionDelDisposition	以库为对象进行撤控
CLIENT_SetGroupInfoForChannel	以通道为对象进行布控

3.4.3 流程说明

以通道或者库为对象进行布控流程，如图 3-3 所示。

图3-3 以通道或者库为对象进行布控流程



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　选择不同的人脸库的布控方式，有以库为对象进行布控，以通道为对象进行布控。

- 以库为对象布控

  1. 选择以库为对象，调用 CLIENT_FaceRecognitionPutDisposition 对库进行布控。

  2. 业务使用完后，调用 CLIENT_FaceRecognitionDelDisposition 对库进行撤控。

- 以通道为对象布控

  选择以通道为对象，调用 CLIENT_SetGroupInfoForChannel 对通道布控。

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

- 以通道为对象或者以库为对象布控都是人脸库布控的方式。
- 以通道为对象布控时，可将多个人脸库部署到一个通道上。以库为对象布控时，可将一

个人脸库部署到多个通道上。

- 以通道为对象布控时，接口 CLIENT_SetGroupInfoForChannel 布控方式为覆盖方式，即总是以最新的布控配置为主。撤控操作为将空的配置布控下去即可。
- 以库为对象撤控时，接口 CLIENT_FaceRecognitionDelDisposition 可对人脸库已经部署的部分通道做撤防，例如：人脸库已经部署在 3 个通道上，撤控的时候可对其部署通道的其中 2 个通道撤控，剩余一个通道保持原样。

3.4.4 示例代码

3.4.4.1 以通道为对象布控

```
// 输入参数

NET_IN_SET_GROUPINFO_FOR_CHANNEL stInChannelDeploy =
{ sizeof(NET_IN_SET_GROUPINFO_FOR_CHANNEL)};

stInChannelDeploy.nChannelID = 0;

stInChannelDeploy.nGroupIdNum = 2; // 需要在该通道配置人脸库的个数

strncpy(stInChannelDeploy.szGroupId[0], strGroupId1, DH_COMMON_STRING_64-1);  // 对人脸库 ID 进行拷贝

strncpy(stInChannelDeploy.szGroupId[1], strGroupId2, DH_COMMON_STRING_64-1);

stInChannelDeploy.nSimilaryNum = 2；// 相似度阈值个数，与人员组数相同

stInChannelDeploy.nSimilary[0] = 85; // 第一个人脸库的人脸相似度阈值

stInChannelDeploy.nSimilary[1] = 90; // 第二个人脸库的人脸相似度阈值


// 输出参数

NET_OUT_SET_GROUPINFO_FOR_CHANNEL stOutChannelDeploy =
{ sizeof(NET_OUT_SET_GROUPINFO_FOR_CHANNEL)};


// 以库为对象布控

BOOL bRet = CLIENT_SetGroupInfoForChannel(lLoginHandle, &stInChannelDeploy,
&stOutChannelDeploy);

if (flase == bRet)

{
```

```
    printf("CLIENT_SetGroupInfoForChannel: failed! Error code: %x.\n", CLIENT_GetLastError());
}

// 以通道为对象撤控，将空的布控信息下发
if (NULL != lRealHandle)
{
memset(stInChannelDeploy, 0, sizeof(NET_IN_SET_GROUPINFO_FOR_CHANNEL));
memset(stOutChannelDeploy, 0, sizeof(NET_OUT_SET_GROUPINFO_FOR_CHANNEL));
stInChannelDeploy.dwSize = sizeof(NET_IN_SET_GROUPINFO_FOR_CHANNEL);
stOutChannelDeploy.dwSize = sizeof(NET_OUT_SET_GROUPINFO_FOR_CHANNEL);
CLIENT_SetGroupInfoForChannel(lLoginHandle, &stInChannelDeploy, &stOutChannelDeploy);
}
```

3.4.4.2 以库为对象布控

```
// 输入参数
NET_IN_FACE_RECOGNITION_PUT_DISPOSITION_INFO stInFaceRecognitionDeploy =
{ sizeof(NET_IN_FACE_RECOGNITION_PUT_DISPOSITION_INFO};
strncpy(stInFaceRecognitionDeploy.szGroupId, strGroupId, DH_COMMON_STRING_64-1); // 需要布控的人脸库
stInFaceRecognitionDeploy.nDispositionChnNum = 2; // 布控的视频通道个数
stInFaceRecognitionDeploy.stuDispositionChnInfo[0].nChannelID = 0;     // 人脸库部署通道
stInFaceRecognitionDeploy.stuDispositionChnInfo[0].nSimilary = 90;     // 相似度阈值
stInFaceRecognitionDeploy.stuDispositionChnInfo[1].nChannelID = 2;
stInFaceRecognitionDeploy.stuDispositionChnInfo[1].nSimilary = 85;

// 输出参数
NET_OUT_FACE_RECOGNITION_PUT_DISPOSITION_INFO stOutFaceRecognitionDeploy =
{sizeof(NET_OUT_FACE_RECOGNITION_PUT_DISPOSITION_INFO)};
```

```
// 以人脸库为对象布控
bool nRet = CLIENT_FaceRecognitionPutDisposition(lLoginHandle, &stInFaceRecognitionDeploy, &stOutFaceRecognitionDeploy);
if(false = nRet)
{
printf("CLIENT_FaceRecognitionPutDisposition: failed! Error code: %x.\n", CLIENT_GetLastError());
}

// 撤控输入参数，可对以部署的部分通道进行人脸库的撤控
NET_IN_FACE_RECOGNITION_DEL_DISPOSITION_INFO stInFaceRecognitionDel = {sizeof(NET_IN_FACE_RECOGNITION_DEL_DISPOSITION_INFO)};
stInFaceRecognitionDel
strncpy(stInFaceRecognitionDel.szGroupId, strGroupId, DH_COMMON_STRING_64-1);
stInFaceRecognitionDel.nDispositionChnNum = 2;  // 撤控的通道个数
stInFaceRecognitionDel.nDispositionChn[0] = 0;  // 通道 1 撤控
stInFaceRecognitionDel.nDispositionChn[1] = 1;

// 撤控输出参数
NET_OUT_FACE_RECOGNITION_DEL_DISPOSITION_INFO stOutFaceRecognitionDel = {sizeof(NET_OUT_FACE_RECOGNITION_DEL_DISPOSITION_INFO)};

bool nRet = CLIENT_FaceRecognitionDelDisposition(lLoginHandle, &stInFaceRecognitionDel, &stOutFaceRecognitionDel);
if(false = nRet)
{
printf("CLIENT_FaceRecognitionDelDisposition: failed! Error code: %x.\n", CLIENT_GetLastError());
}
```

3.5 以图搜图

3.5.1 简介

外部导入一张图片和相似度值，IVSS、NVR 等设备通过这张图检索历史库或人脸库是否已经存在匹配的人脸，并返回在相似度之上的图片结果。

3.5.2 接口总览

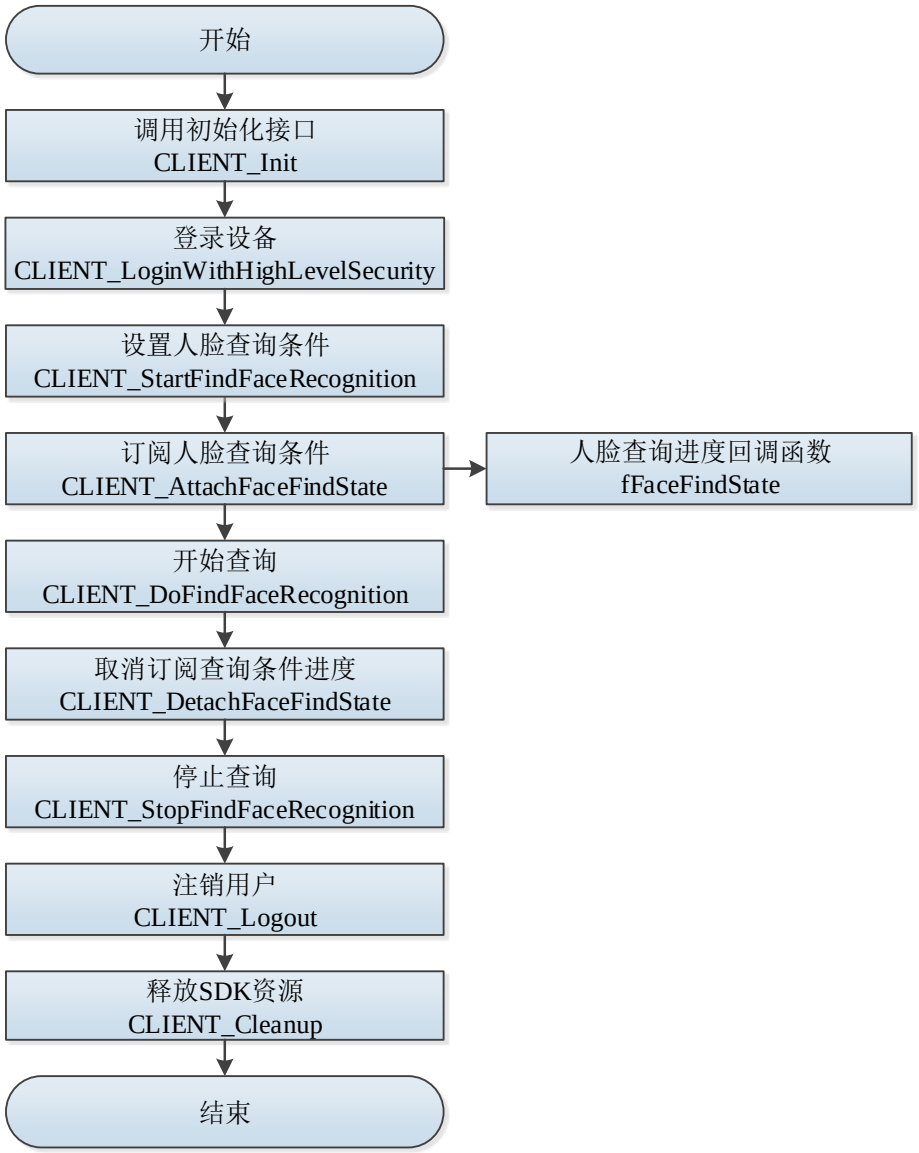
表3-4 以图搜图接口说明

接口	说明
CLIENT_StartFindFaceRecognition	设置人脸的查询条件
CLIENT_AttachFaceFindState	订阅人脸查询条件
CLIENT_DetachFaceFindState	取消订阅查询条件进度
CLIENT_DoFindFaceRecognition	开始查询
CLIENT_StopFindFaceRecognition	结束查询

3.5.3 流程说明

以图搜图流程，如图 3-4 所示。

图3-4 以图搜图流程



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_StartFindFaceRecognition 设置人脸查询条件。

步骤4　查看步骤 3 中的返回值，若接口出参结构体中 nTotalCount 返回的是-1，需要等待设备查询完成。

步骤5　调用 CLIENT_AttachFaceFindState 订阅人脸查询状态。然后等待进度回调函数中的返回进度为 100 时即搜索完成。搜索完成后调用 CLIENT_DetachFaceFindState 取消订阅查询进度。

步骤6　调用 CLIENT_DoFindFaceRecognition 接口获取查询结果。

步骤7　调用 CLIENT_StopFindFaceRecognition 接口结束查询。

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。

3.5.4 示例代码

```
// 查询进度回调函数
void CALLBACK FaceFindState(LLONG lLoginID, LLONG lAttachHandle, NET_CB_FACE_FIND_STATE*
pstStates, int nStateNum, LDWORD dwUser)
{
if (pstStates->nProgress== 100) //表示查询进度 100%
{
// 停止人脸查询进度的订阅
CLIENT_DetachFaceFindState(lAttachHandle);
// 开始查询
DoFind();
}
return;
}

// 配置查询条件
NET_IN_STARTFIND_FACERECONGNITION stuInParam = { sizeof(stuInParam) };
NET_OUT_STARTFIND_FACERECONGNITION stuOutParam = { sizeof(stuOutParam) };
stuInParam.stFilterInfo.dwSize = sizeof(stuInParam.stFilterInfo);
stuInParam.stMatchOptions.dwSize = sizeof(stuInParam.stMatchOptions);
stuInParam.bPersonExEnable = TRUE;
stuInParam.nChannelID = 0;
stuInParam.stMatchOptions.nSimilarity = 80;
stuInParam.stFilterInfo.stStartTime = startTime;
stuInParam.stFilterInfo.stEndTime = endTime;
```

```
stuInParam.nBufferLen = nPicBufLen;
stuInParam.pBuffer = strPicBuf;    // 图片 Buffer
stuInParam.stPersonInfoEx.wFacePicNum = 1;
stuInParam.stPersonInfoEx.szFacePicInfo[0].dwOffSet = 0;
stuInParam.stPersonInfoEx.szFacePicInfo[0].dwFileLenth = nLength;

BOOL bRet = CLIENT_StartFindFaceRecognition(m_lLoginId, &stuInParam, &stuOutParam, 5000);
if (!bRet)
{
printf("CLIENT_StartFindFaceRecognition: failed! Error code %x.\n", CLIENT_GetLastError());
return -1;
}
m_lFindHandle = stuOutParam.lFindHandle;

if (-1 == stuOutParam.nTotalCount)
{
// 当查询总数为-1 时，表示设备未查询完毕，需要订阅设备查询进度
NET_IN_FACE_FIND_STATE stuInFindState = { sizeof(stuInFindState) };
NET_OUT_FACE_FIND_STATE stuOutFindState = { sizeof(stuOutFindState) };
stuInFindState.nTokenNum = 1;
int nToken = stuOutParam.nToken;
stuInFindState.nTokens = &nToken;
stuInFindState.cbFaceFindState = FaceFindState; // 进度回调函数
stuInFindState.dwUser = (DWORD)this;
m_lAttachHandle = CLIENT_AttachFaceFindState(m_lLoginId, &stuInFindState, &stuOutFindState,
5000);
}
else
{
```

```
// 直接开始查询
DoFind();
}

void DoFind()
{
NET_IN_DOFIND_FACERECONGNITION stuInDoFind =
{ sizeof(NET_IN_DOFIND_FACERECONGNITION) };
NET_OUT_DOFIND_FACERECONGNITION stuOutDoFind =
{ sizeof(NET_OUT_DOFIND_FACERECONGNITION) };
stuOutDoFind.bUseCandidatesEx = TRUE;
stuInDoFind.nCount = 20; // 每次查询 20 条信息,本次查询的总数大于 20
stuInDoFind.lFindHandle = m_lFindHandle;
stuInDoFind.emDataType = EM_NEEDED_PIC_TYPE_HTTP_URL;// 指定查询结果返回图片的格式为 http 链接
stuInDoFind.nBeginNum = 0;  // 从 0 开始查询

BOOL bRet = CLIENT_DoFindFaceRecognition(&stuInDoFind, &stuOutDoFind, 10000);
if (!bRet)
{
printf("CLIENT_DoFindFaceRecognition: failed! Error code %x.\n", CLIENT_GetLastError());
return ;
}

CLIENT_StopFindFaceRecognition(m_lFindHandle);
}
```

3.6 人脸相关录像图片查询与下载

因为人脸相关的智能为智能事件的一种，因此人脸相关的录像图片查询请参见“2.5 查询/回放/下载录像和图片”。

在查询时，调用接口 CLIENT_FindFileEx 设置查询条件，关于人脸相关的查询操作，接口参数 emType 的取值情况如下：

- DH_FILE_QUERY_FACE 查询人脸图片
- DH_FILE_QUERY_FACE_DETECTION 查询目标检测图片
- DH_FILE_QUERY_FILE 查询录像，且接口参数 pQueryCondition 为类型

NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：

- EVENT_IVS_FACERECOGNITION 目标识别录像查询
- EVENT_IVS_FACEDETECT 目标检测录像查询

第 4 章 人体检测

4.1 人体事件订阅

具体的订阅请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出人体检测事件，即 EVENT_IVS_HUMANTRAIT 人体检测事件。

4.2 人体图片搜索

4.2.1 简介

人体检测的图片搜索的代码参考“2.5 查询/回放/下载录像和图片”中的图片搜索，以下为人体检测的图片下载的介绍。

4.2.2 接口总览

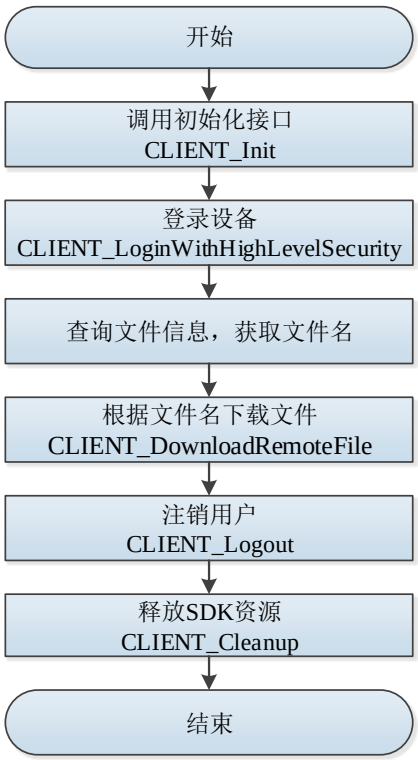
表4-1 人体图片下载接口说明

接口	说明
CLIENT_DownloadRemoteFile	下载文件

4.2.3 流程说明

人体图片搜索流程，如图 4-1 所示。

图4-1 人体图片搜索流程



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　调用 SDK 接口查询文件信息，获取需要下载文件的文件名。

步骤4　使用查询到的文件信息，调用接口 CLIENT_DownloadRemoteFile 下载文件。

步骤5　业务使用完后，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。

4.2.4 示例代码

```
DH_IN_DOWNLOAD_REMOTE_FILE stuInDownloadFile    =
{sizeof(DH_IN_DOWNLOAD_REMOTE_FILE    )};

stuInDownloadFile.pszFileName = strFileName

stuInDownloadFile.pszFileDst = strDownloadName;



DH_OUT_DOWNLOAD_REMOTE_FILE stuOutDownloadFile =
{sizeof(DH_OUT_DOWNLOAD_REMOTE_FILE )};
```

```
BOOL bRet = CLIENT_DownloadRemoteFile(m_lLoginHandle, &stuInDownloadFile,
&stuOutDownloadFile);

if (bRet == FALSE)

{

    MessageBox(ConvertString("Download Failed"), ConvertString("Prompt"));

    return;

}
```

第 5 章 客流量统计

5.1 客流量事件订阅

5.1.1 简介

客流量统计功能指在经营区域安装前端设备，智能分析服务器根据前端采集的视频数据精确统计出每个入口实时客流进出人数。该类产品被广泛应用于大型商业、旅游行业、公共安全、文博和连锁等行业。

客流量数据的实时订阅，可以获取到当天总出入人数情况和实时出入的人数的统计情况。

5.1.2 接口总览

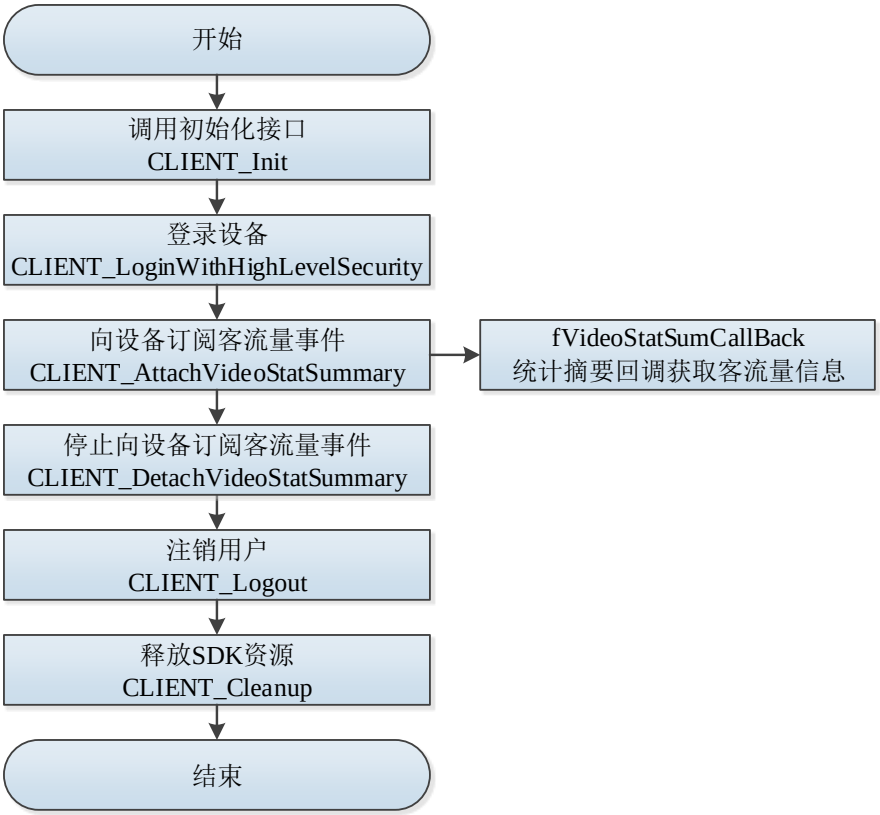
表5-1 客流量事件订阅接口说明

接口	说明
CLIENT_AttachVideoStatSummary	订阅客流量事件
CLIENT_DetachVideoStatSummary	退订客流量事件

5.1.3 流程说明

客流量订阅流程，如图 5-1 所示。

图5-1 客流量订阅流程



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_AttachVideoStatSummary 向设备订阅客流量事件。

步骤4　订阅成功后，设备上报的客流量事件通过 fVideoStatSumCallBack 回调函数获取人脸事件并通知给用户。

步骤5　客流量事件上报功能使用完毕后，调用 CLIENT_DetachVideoStatSummary 停止订阅客流量事件。

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。

5.1.4 示例代码

```
void CALLBACK VideoStatSumCallback(LLONG lAttachHandle, NET_VIDEOSTAT_SUMMARY* pBuf,
DWORD dwBufLen, LDWORD dwUser)

{

    // 处理回调数据
```

```
}


NET_IN_ATTACH_VIDEOSTAT_SUM InParam = {sizeof(NET_IN_ATTACH_VIDEOSTAT_SUM)};
NET_OUT_ATTACH_VIDEOSTAT_SUM OutParam = {sizeof(NET_OUT_ATTACH_VIDEOSTAT_SUM)};
InParam.nChannel=0;
InParam.cbVideoStatSum=VideoStatSumCallback;   // 订阅回调函数


// 订阅客流量统计
LLONG attachHnd = CLIENT_AttachVideoStatSummary(lLoginID,&InParam,&OutParam,5000)
if(0 == attachHnd)
{
    printf("CLIENT_AttachVideoStatSummary failed! Error code %x.\n", CLIENT_GetLastError());
    return;
}


// 取消订阅客流量统计
CLIENT_DetachVideoStatSummary(attachHnd);
```

5.2 客流量事件报警

具体的订阅请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出客流量智能事件，即“EVENT_IVS_NUMBERSTAT 人数统计事件”和“EVENT_IVS_MAN_NUM_DETECTION 区域人数统计事件”。

5.3 客流量历史数据查询

5.3.1 简介

用户指定客流量信息的起始时间和结束时间，设备端将查询结果返回给 SDK。

5.3.2 接口总览

表5-2 客流量历史数据查询接口说明

接口	说明
CLIENT_StartFindNumberStat	开始查询历史客流量信息
CLIENT_DoFindNumberStat	继续查询历史客流量信息
CLIENT_StopFindNumberStat	结束查询历史客流量信息

5.3.3 流程说明

客流图片录像的查询流程，如图 5-2 所示。

图5-2 录像或图片的查询流程



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_StartFindNumberStat 开始获取人数统计信息。

步骤4　调用 CLIENT_DoFindNumberStat 继续查询某段时间的客流量统计信息。

步骤5　调用 CLIENT_StopFindNumberStat 停止记录查询。

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。

5.3.4 示例代码

```
// 设置查询条件
NET_IN_FINDNUMBERSTAT inParam = { sizeof(NET_IN_FINDNUMBERSTAT)};
inParam.nChannelID = nChannelID; // 要进行查询的通道号
inParam.nGranularityType = 1; // 查询粒度 0：分钟，1：小时，2：日，3：周，4：月，5：季，6：年
inParam.nWaittime = 5000; // 等待接收数据的超时时间
NET_OUT_FINDNUMBERSTAT outParam { sizeof(NET_OUT_FINDNUMBERSTAT)};
LLONG findHnd = CLIENT_StartFindNumberStat(pLoginHandle, &inParam, &outParam);
if (findHand == 0)
{
printf("CLIENT_StartFindNumberStat failed! Error code %x.\n", CLIENT_GetLastError());
return ;
}

NET_IN_DOFINDNUMBERSTAT inDoFind = {sizeof(NET_IN_DOFINDNUMBERSTAT)};
NET_OUT_DOFINDNUMBERSTAT outDoFind = {sizeof(NET_OUT_DOFINDNUMBERSTAT)};
inDoFind.nBeginNumber = 0; // 从 0 开始查询
inDoFind.nCount = 10; // 每次查询 10 条
inDoFind.nWaittime = 5000; // 接口超时时间 5s

outDoFind.pstuNumberStat = new DH_NUMBERSTAT[10];
outDoFind.nBufferLen = 10 * sizeof(DH_NUMBERSTAT);
```

```
for (int i = 0; i < 10 ; i++)
{
    outDoFind.pstuNumberStat[i].dwSize = sizeof(DH_NUMBERSTAT);
}

// 查询
BOOL bRet = CLIENT_DoFindNumberStat(findHand, &inDoFind, &outDoFind)
if (FALSE == bRet)
{
printf("CLIENT_DoFindNumberStat failed! Error code %x.\n", CLIENT_GetLastError());
delete[] outDoFind.pstuNumberStat;
return ;
}

// 停止查询客流量
CLINET_StopFindNumberStat(findHand);
delete[] outDoFind.pstuNumberStat;
```

第 6 章 通用行为事件

6.1 通用行为事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

- EVENT_IVS_CROSSLINEDETECTION 绊线入侵事件
- EVENT_IVS_CROSSREGIONDETECTION 区域入侵事件
- EVENT_IVS_CROSSFENCEDETECTION 穿越围栏报警事件
- EVENT_IVS_LEFTDETECTION 物品遗留报警事件
- EVENT_IVS_MOVEDETECTION 快速移动报警事件
- EVENT_IVS_RIOTERDETECTION 人员聚集报警事件
- EVENT_IVS_TAKENAWAYDETECTION 物品搬移报警事件
- EVENT_IVS_PARKINGDETECTION 停车检测报警事件
- EVENT_IVS_WANDERDETECTION 徘徊检测报警事件

6.2 通用行为事件录像查询和下载

通用行为为智能事件的一种，故相关录像和图片查询信息请参见“2.5 查询/回放/下载录像和图片”。

调用接口 CLIENT_FindFileEx 设置查询条件。通用行为录像的查询操作、接口参数如下：

- emType：DH_FILE_QUERY_FILE 查询录像。
- pQueryCondition：NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：
  - ◇ EVENT_IVS_CROSSLINEDETECTION 绊线入侵录像查询。
  - ◇ EVENT_IVS_CROSSREGIONDETECTION 区域入侵录像查询。
  - ◇ EVENT_IVS_CROSSFENCEDETECTION 穿越围栏报警录像查询。
  - ◇ EVENT_IVS_LEFTDETECTION 物品遗留报警录像查询。
  - ◇ EVENT_IVS_MOVEDETECTION 快速移动报警录像查询。
  - ◇ EVENT_IVS_RIOTERDETECTION 人员聚集报警录像查询。
  - ◇ EVENT_IVS_TAKENAWAYDETECTION 物品搬移报警录像查询。

◇ EVENT_IVS_PARKINGDETECTION 停车检测报警录像查询。

◇ EVENT_IVS_WANDERDETECTION 徘徊检测报警录像查询。

第 7 章 智能交通

7.1 智能交通事件订阅

具体的订阅请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出智能交通事件，如下：

- EVENT_IVS_TRAFFICJUNCTION 交通路口事件
- EVENT_IVS_TRAFFICJAM 交通拥堵事件
- EVENT_IVS_TRAFFIC_OVERSPEED 超速事件
- EVENT_IVS_TRAFFIC_UNDERSPEED 欠速
- EVENT_IVS_TRAFFIC_PEDESTRAIN 行人事件
- EVENT_IVS_TRAFFIC_FLOWSTATE 车流量事件
- EVENT_IVS_TRAFFIC_VEHICLEINROUTE 有车占道
- EVENT_IVS_TRAFFIC_NON_MOTOR_RETROGRADE 非机动车逆行
- EVENT_IVS_TRAFFIC_OVERLINE 压白线
- EVENT_IVS_TRAFFIC_OVERYELLOWLINE 压黄线
- EVENT_IVS_TRAFFIC_RETROGRADE 逆行
- EVENT_IVS_TRAFFIC_CROSSLANE 违法变道
- EVENT_IVS_TRAFFIC_QUEUEJUMP 车辆排队加塞
- EVENT_IVS_TRAFFIC_YELLOWPLATEINLANE 黄牌占道
- EVENT_IVS_TRAFFIC_WRONGROUTE 不按车道行驶
- EVENT_IVS_TRAFFIC_UNDERSPEED 低速
- EVENT_IVS_TRAFFIC_OVERSPEED 超速
- EVENT_IVS_TRAFFIC_TRUCKFORBID 禁货
- EVENT_IVS_TRAFFIC_UTURN 违法掉头
- EVENT_IVS_TRAFFIC_TURNLEFT 违法左转
- EVENT_IVS_TRAFFIC_TURNRIGHT 违法右转
- EVENT_IVS_TRAFFIC_BACKING 违法倒车
- EVENT_IVS_TRAFFIC_PEDESTRAINPRIORITY 未礼让行人

7.2 车流量历史数据查询

7.2.1 简介

查询车流量的历史数据。

7.2.2 接口总览

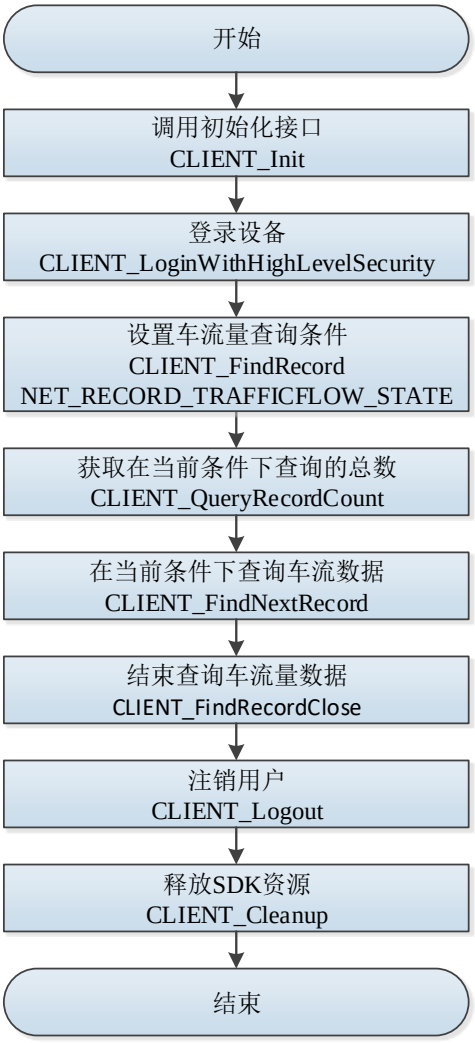
表7-1 车流量的历史数据查询接口说明

接口	说明
CLIENT_FindRecord	设置查询条件
CLIENT_QueryRecordCount	获取查询数量
CLIENT_FindNextRecord	在当前查询条件下查询数据
CLIENT_FindRecordClose	关闭查询

7.2.3 流程说明

车流量历史数据查询流程，如图 7-1 所示。

图7-1 车流量历史数据查询



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　调用接口 CLIENT_FindRecord 设置车流量的查询条件，使用的枚举为 NET_RECORD_TRAFFICFLOW_STATE。

步骤4　调用接口 CLIENT_QueryRecordCount，获取在当前查询条件下可以查到的数据总数。

步骤5　调用接口 CLIENT_FindNextRecord，在当前查询条件下查询指定条数的车流数据。

步骤6　查询结束后，调用接口 CLIENT_FindRecordClose 清理查询资源。

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

- 在查询车流量过程中，首先，设备必须支持该功能，且在查询时间段内有车流数据。此外，需要设备有 SD 卡，这样可以保存车流数据。
- 在查询的车流中，平均速度字段如果为-1，则表示该时间段没有车辆经过；如果大于 0，则表示车辆的平均速度；如果为 0，则表示平均速度为 0。

7.2.4 示例代码

```
// 开始查询，设置查询条件

FIND_RECORD_TRAFFICFLOW_CONDITION stTrafficFlow =
{sizeof(FIND_RECORD_TRAFFICFLOW_CONDITION)};

stTrafficFlow.abChannelId =TRUE;

stTrafficFlow.nChannelId = 0;

stTrafficFlow.abLane = FALSE;

stTrafficFlow.bStartTime= TRUE;

stTrafficFlow.bEndTime= TRUE;

stTrafficFlow.stStartTime = startTime;

stTrafficFlow.stEndTime = endTime;

stTrafficFlow.bStatisticsTime = TRUE;


NET_IN_FIND_RECORD_PARAM stuFindInParam = {sizeof(NET_IN_FIND_RECORD_PARAM)};

stuFindInParam.emType = NET_RECORD_TRAFFICFLOW_STATE;

stuFindInParam.pQueryCondition = &stTrafficFlow;


NET_OUT_FIND_RECORD_PARAM stuFindOutParam = {sizeof(NET_OUT_FIND_RECORD_PARAM)};

bool bRet = CLIENT_FindRecord(m_lLoginHandle, &stuFindInParam, &stuFindOutParam,
MAX_TIMEOUT);

if (!bRet)

{

return;
```

```
}


// 查询总数
NET_IN_QUEYT_RECORD_COUNT_PARAM inQueryCountParam =
{ sizeof(NET_IN_QUEYT_RECORD_COUNT_PARAM)};
inQueryCountParam.lFindeHandle =    stuFindOutParam.lFindeHandle;
NET_OUT_QUEYT_RECORD_COUNT_PARAM outQueryCountParam    =
{ sizeof(NET_OUT_QUEYT_RECORD_COUNT_PARAM) };


bRet = CLIENT_QueryRecordCount(&inQueryCountParam, &outQueryCountParam ,
MAX_TIMEOUT);
if (!bRet)
{
MessageBox(ConvertString("Query record count failed!"), ConvertString("Prompt"));
return;
}



// 查询 100 条数据
int nQueryCount = 100;
NET_RECORD_TRAFFIC_FLOW_STATE* pRecordList = new
NET_RECORD_TRAFFIC_FLOW_STATE[nQueryCount];
memset(pRecordList, 0, sizeof(NET_RECORD_TRAFFIC_FLOW_STATE) * nQueryCount);
for (int unIndex = 0; unIndex < nQueryCount; ++unIndex)
{
pRecordList[unIndex].dwSize = sizeof(NET_RECORD_TRAFFIC_FLOW_STATE);
}


NET_IN_FIND_NEXT_RECORD_PARAM stuFindNextInParam =
{sizeof(NET_IN_FIND_NEXT_RECORD_PARAM)};
```

```
stuFindNextInParam.lFindeHandle = stuFindOutParam.lFindeHandle;

stuFindNextInParam.nFileCount = nQueryCount;



NET_OUT_FIND_NEXT_RECORD_PARAM stuFindNextOutParam =
{sizeof(NET_OUT_FIND_NEXT_RECORD_PARAM)};

stuFindNextOutParam.pRecordList = pRecordList;

stuFindNextOutParam.nMaxRecordNum = nQueryCount;

bRet = CLIENT_FindNextRecord(&stuFindNextInParam, &stuFindNextOutParam, MAX_TIMEOUT);

if (!bRet)

{

MessageBox(ConvertString("Query record count failed!"), ConvertString("Prompt"));

}



// 结束查询

CLIENT_FindRecordClose(stuFindOutParam.lFindeHandle);

delete[] pRecordList;
```

7.3 车辆黑白名单增删改查

7.3.1 简介

对车辆的黑白名单进行增删改查操作。

7.3.2 接口总览

表7-2 车辆黑白名单增删改查相关接口说明

接口	说明
CLIENT_OperateTrafficList	黑白名单的增删改操作
CLIENT_FindRecord	设置查询条件
CLIENT_QueryRecordCount	获取查询数量
CLIENT_FindNextRecord	在当前查询条件下查询数据

接口	说明
CLIENT_FindRecordClose	关闭查询

7.3.3 流程说明

车辆黑白名单增删改查流程，如图 7-2 所示。

图7-2 车辆黑白名单增删改查



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 调用接口 CLIENT_OperateTrafficList，枚举类型为 NET_TRAFFIC_LIST_INSERT，增加黑白名单。

步骤4 调用接口 CLIENT_FindRecord 设置黑白名单的查询条件，使用的枚举为 NET_RECORD_TRAFFICREDLIST(白名单)、NET_RECORD_TRAFFICBLACKLIST（黑名单）。

步骤5 调用接口 CLIENT_QueryRecordCount，获取在当前查询条件下可以查到的数据总数。

步骤6 调用接口 CLIENT_FindNextRecord，在当前查询条件下查询指定条数的黑白名单数量。

步骤7 使用查询到的黑白名单信息， 调用接口 CLIENT_OperateTrafficList 进行黑白名单的修改删除操作，枚举为 NET_TRAFFIC_LIST_REMOVE（删除）、NET_TRAFFIC_LIST_UPDATE（修改）。

步骤8 查询结束后，调用接口 CLIENT_FindRecordClose 清理查询资源。

步骤9 业务使用完后，调用 CLIENT_Logout 登出设备。

步骤10 SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

调用接口 CLIENT_OperateTrafficList 增加黑白名单的操作时，当接口返回-1，不表示接口失败，而是表示还未生成记录集号。因为设备将增加的名单先保存到缓冲中，就返回添加成功的结果，之后再将缓冲的数据保存到数据库中。在设备返回结果的时候，因为名单还未添加到数据库中，因此没有记录集编号，设备那边就返回-1 表示。

7.3.4 示例代码

查询黑白名单的代码请参见“7.2.4 示例代码”，以下为黑白名单的增删改查的示例代码。

```
// 黑白名单的增加

NET_IN_OPERATE_TRAFFIC_LIST_RECORD stInParam =
{ sizeof(NET_IN_OPERATE_TRAFFIC_LIST_RECORD) };

NET_OUT_OPERATE_TRAFFIC_LIST_RECORD stOutParam =
{ sizeof(NET_OUT_OPERATE_TRAFFIC_LIST_RECORD) };

stInParam.emOperateType = NET_TRAFFIC_LIST_INSERT;

stInParam.emRecordType = NET_RECORD_TRAFFICBLACKLIST; // 黑名单

//stInParam.emRecordType = NET_RECORD_TRAFFICREDLIST; // 白名单
```

```
NET_TRAFFIC_LIST_RECORD stTrafficListRecord = { sizeof(NET_TRAFFIC_LIST_RECORD) };
stTrafficListRecord.stBeginTime = startTime;
stTrafficListRecord.stCancelTime = endTime;
strncpy(stTrafficListRecord.szPlateNumber, strPlateNumber.GetBuffer(),
DH_MAX_PLATE_NUMBER_LEN-1);
strncpy(stTrafficListRecord.szMasterOfCar, strOwner.GetBuffer(), DH_MAX_NAME_LEN-1);

NET_INSERT_RECORD_INFO stInsertInfo = { sizeof( NET_INSERT_RECORD_INFO ) };
stInsertInfo.pRecordInfo = &stTrafficListRecord;
stInParam.pstOpreateInfo = &stInsertInfo;



bool bRet = CLIENT_OperateTrafficList(m_lLoginHandle, &stInParam, &stOutParam,
MAX_TIMEOUT);
if (!bRet)
{
return;
}

// 黑白名单的修改
NET_IN_OPERATE_TRAFFIC_LIST_RECORD stInParam =
{ sizeof(NET_IN_OPERATE_TRAFFIC_LIST_RECORD) };
NET_OUT_OPERATE_TRAFFIC_LIST_RECORD stOutParam =
{ sizeof(NET_OUT_OPERATE_TRAFFIC_LIST_RECORD) };

stInParam.emOperateType = NET_TRAFFIC_LIST_UPDATE;
stInParam.emRecordType = NET_RECORD_TRAFFICBLACKLIST; // 黑名单
//stInParam.emRecordType = NET_RECORD_TRAFFICREDLIST; // 白名单
```

```
NET_TRAFFIC_LIST_RECORD stTrafficListRecord = { sizeof(NET_TRAFFIC_LIST_RECORD) };

stTrafficListRecord.stBeginTime = startTime;

stTrafficListRecord.stCancelTime = endTime;

strncpy(stTrafficListRecord.szPlateNumber, strPlateNumber.GetBuffer(),
DH_MAX_PLATE_NUMBER_LEN-1);

strncpy(stTrafficListRecord.szMasterOfCar, strOwner.GetBuffer(), DH_MAX_NAME_LEN-1);

stTrafficListRecord.nRecordNo = m_stTrafficListInfo.nRecordNo;    // 记录集编号


NET_UPDATE_RECORD_INFO stModifyRecord = {sizeof(NET_UPDATE_RECORD_INFO)};

stModifyRecord.pRecordInfo = &stTrafficListRecord;

stInParam.pstOpreateInfo = &stModifyRecord;


bool bRet = CLIENT_OperateTrafficList(m_lLoginHandle, &stInParam, &stOutParam,
MAX_TIMEOUT);

if (!bRet)

{

      return;

}


// 黑白名单的删除

NET_REMOVE_RECORD_INFO stRemoveRecord = { sizeof(NET_REMOVE_RECORD_INFO) };

stRemoveRecord.nRecordNo = m_vecTrafficListInfo[nSelect]->nRecordNo; // 记录集编号


NET_IN_OPERATE_TRAFFIC_LIST_RECORD stInParam =
{ sizeof(NET_IN_OPERATE_TRAFFIC_LIST_RECORD) };

stInParam.emOperateType = NET_TRAFFIC_LIST_REMOVE;

stInParam.emRecordType = NET_RECORD_TRAFFICBLACKLIST;    // 黑名单

//stInParam.emRecordType = NET_RECORD_TRAFFICREDLIST; // 白名单
```

```
stInParam.pstOpreateInfo = &stRemoveRecord;

NET_OUT_OPERATE_TRAFFIC_LIST_RECORD stOutParam =
{ sizeof(NET_OUT_OPERATE_TRAFFIC_LIST_RECORD) };



bool bRet = CLIENT_OperateTrafficList(m_lLoginHandle, &stInParam, &stOutParam,
MAX_TIMEOUT);

if (!bRet)

{

return;

}
```

7.4 车辆相关图片的查询与下载

7.4.1 简介

智能交通中会将智能事件的抓拍图片保存到设备的存储器上，可使用车牌或者相应智能事件等来搜索对应的图片，并支持对图片的下载。

车辆相关图片搜索请参见“2.5 查询/回放/下载录像和图片”中的图片查询内容，且接口 CLIENT_FindFileEx 查询的类型为 DH_FILE_QUERY_TRAFFICCAR_EX。

以下为查询到图片下载使用的接口及示例代码。

7.4.2 接口总览

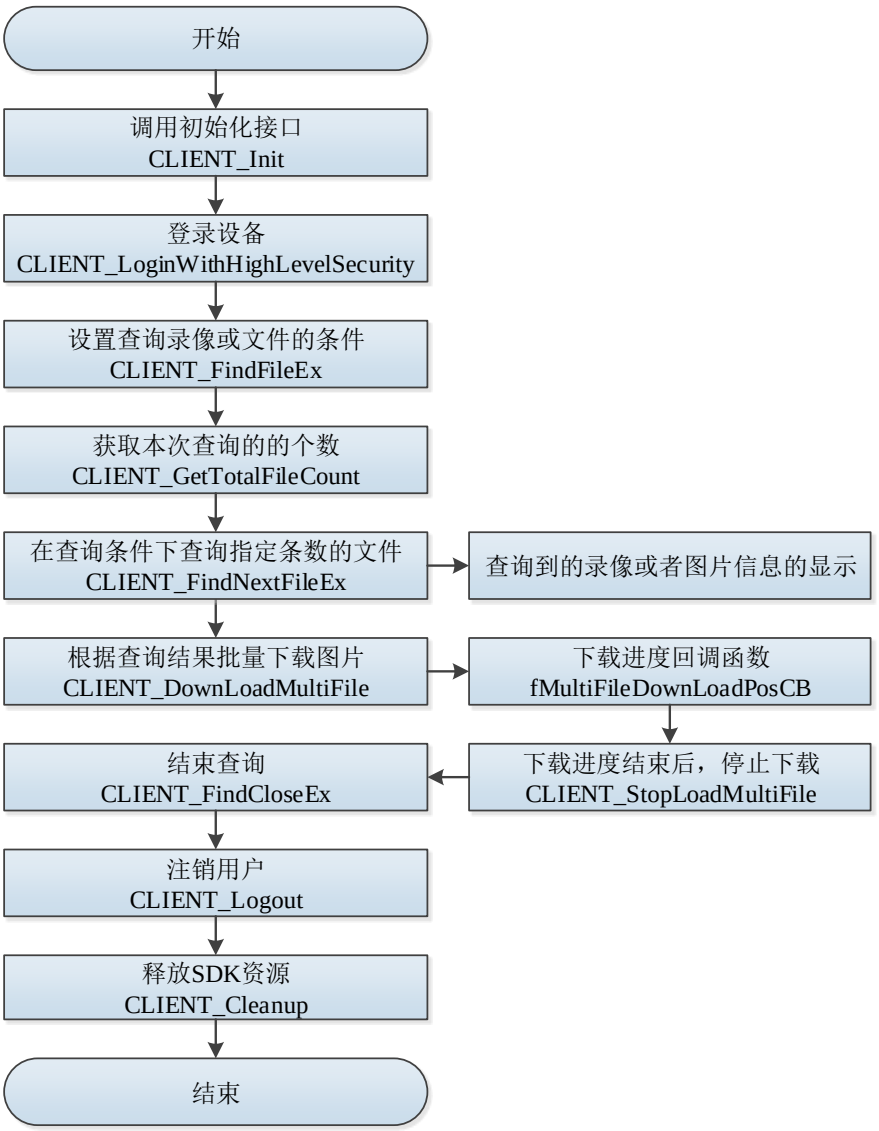
表7-3 车辆相关图片下载相关接口说明

接口	说明
CLIENT_DownLoadMultiFile	批量下载文件
CLIENT_StopLoadMultiFile	停止批量下载文件

7.4.3 流程说明

图片查询与下载流程，如图 7-3 所示。

图7-3 图片查询与下载过程



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　调用 CLIENT_FindFileEx 设置查询条件，设置成功后返回查询句柄。根据 emType 的不同取值，判断查找的类型。

步骤4　调用 CLIENT_GetTotalFileCount 获取查询到的录像或者文件总数。

步骤5　调用 CLIENT_FindNextFileEx 查询指定条数的录像或者图片，并将查询到的信息保存，用于录像或者图片的回放下载操作。

步骤6　使用查询到的文件信息，调用接口 CLIENT_DownLoadMultiFile 批量下载文件。

步骤7　下载进度回调函数 fMultiFileDownLoadPosCB 中的参数 dwDownLoadSize 为最大值时，调用接口 CLIENT_StopLoadMultiFile 结束图片下载。

步骤8　调用 CLIENT_FindCloseEx 结束查询。

步骤9　业务使用完后，调用 CLIENT_Logout 登出设备。

步骤10 SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 接口 CLIENT_FindFileEx 中参数 pQueryCondition 是由用户申请和释放的，具体类型由 emType 的枚举类型来定义。
- CLIENT_FindFileEx 如果查询成功，则会返回查询句柄，CLIENT_FindNextFileEx 将查询句柄作为参数查询具体的录像或者图片，且必须调用 CLIENT_FindCloseEx 将查询句柄关闭。
- CLIENT_FindNextFileEx 可设置查询条数，如果设置条数大于 1 条则参数 pMediaFileInfo 必须为一个数据指针，数据大小大于等于设置的条数。

7.4.4 示例代码

```
// 批量下载进度回调函数

void CALLBACK DownloadTrafficPicture(LLONG lDownLoadHandle, DWORD dwID, DWORD
dwFileTotalSize, DWORD dwDownLoadSize, int nError, LDWORD dwUser, void* pReserved)

{

if (nError != 0 || dwDownLoadSize == UINT_MAX)

{

// 下载出错或者下载结束

CLIENT_StopLoadMultiFile(m_lDownloadHandle);

delete[] pDownLoadInfo;

}

}



NET_DOWNLOADFILE_INFO* pDownLoadInfo = new NET_DOWNLOADFILE_INFO[10]; //下载 10
文件

memset(pDownLoadInfo, 0, 10*sizeof(NET_DOWNLOADFILE_INFO));

for(int i =0; i++; i<10)

{
```

```
pDownLoadInfo[i].dwFileID = 1;


// stTrafficPicture 的数据是通过 CLIENT_FindFileEx 查询出来的文件信息中的文件，类型为 MEDIAFILE_TRAFFICCAR_INFO_EX

pDownLoadInfo[i].nFileSize = stTrafficPicture.stuInfo.sizeEx / 1024;

strncpy(pDownLoadInfo[i].szSourceFilePath, stTrafficPicture.stuInfo.szFilePath, MAX_PATH-1);


// 下载后的文件保存路径

strncpy(pDownLoadInfo[i].szSavedFileName, szFilePathName[i], MAX_PATH-1);

}


NET_IN_DOWNLOAD_MULTI_FILE stInDownloadFile = {sizeof(NET_IN_DOWNLOAD_MULTI_FILE )};

NET_OUT_DOWNLOAD_MULTI_FILE stOutDownloadFile=
{sizeof(NET_OUT_DOWNLOAD_MULTI_FILE )};

stInDownloadFile.emDownloadType = EM_DOWNLOAD_BY_FILENAME;

stInDownloadFile.cbPosCallBack = DownloadTrafficPicture;   // 批量下载进度回调函数

stInDownloadFile.dwUserData =   (LDWORD)this;

stInDownloadFile.nFileCount = 10;   // 下载文件个数

stInDownloadFile.pFileInfos = pDownLoadInfo; // 需要下载的文件信息指针，如需下载多个则为数组首地址


// 批量下载图片文件

BOOL nRet = CLIENT_DownLoadMultiFile(m_lLoginHandle, &stInDownloadFile,
&stOutDownloadFile, MAX_TIMEOUT);

if (!nRet)

{

printf("CLIENT_DownLoadMultiFile failed! Error code %x.\n", CLIENT_GetLastError());

delete[] pDownLoadInfo;

return;

}
```

```
m_lDownloadHandle = stOutDownloadFile.lDownLoadHandle;
```

第 8 章 闸机

8.1 门禁事件订阅

具体的订阅请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出门禁事件，如下：

EVENT_IVS_ACCESS_CTL 门禁事件

8.2 门禁卡信息管理

8.2.1 简介

对门禁卡信息如卡号、用户 ID、卡名等相关信息进行增删改查的操作。

8.2.2 接口总览

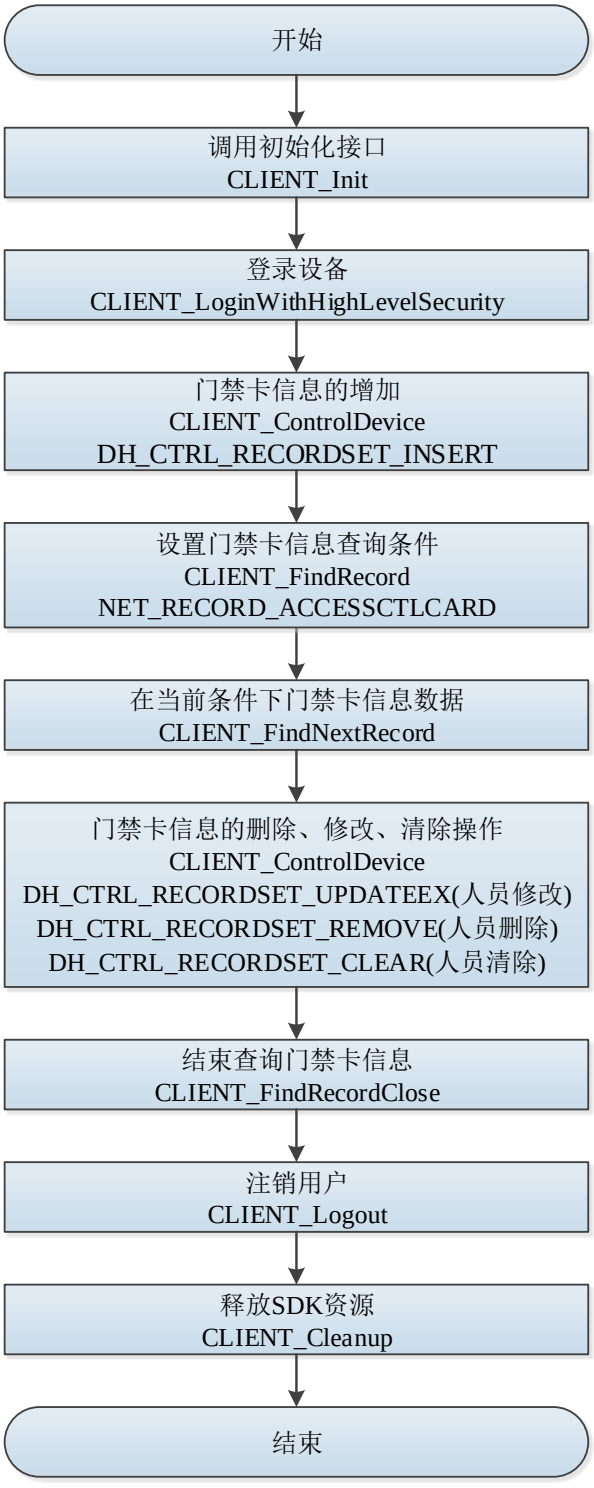
表8-1 门禁卡信息管理接口说明

接口	说明
CLIENT_FindRecord	设置查询条件
CLIENT_FindNextRecord	在当前查询条件下查询数据
CLIENT_FindRecordClose	关闭查询
CLIENT_ControlDevice	门禁卡信息的增加、修改、删除、清除

8.2.3 流程说明

门禁卡信息的增删改查流程，如图 8-1 所示。

图8-1 门禁卡信息的增删改查



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_ControlDevice 增加门禁卡信息，使用的枚举为 DH_CTRL_RECORDSET_INSERT。

步骤4 调用接口 CLIENT_FindRecord 设置门禁卡信息的查询条件，使用的枚举为 NET_RECORD_ACCESSCTLCARD。

步骤5 调用接口 CLIENT_FindNextRecord，在当前查询条件下查询指定条数的门禁卡信息数据。

步骤6 查询结束后，调用接口 CLIENT_FindRecordClose 清理查询资源。

步骤7 调用 CLIENT_ControlDevice，修改门禁卡信息，使用的枚举为 NET_RECORD_ACCESSCTLCARD。

步骤8 调用 CLIENT_ControlDevice，删除门禁卡信息，使用的枚举为 DH_CTRL_RECORDSET_REMOVE。

步骤9 调用 CLIENT_ControlDevice，清除门禁卡信息，使用的枚举为 DH_CTRL_RECORDSET_CLEAR。

步骤10 业务使用完后，调用 CLIENT_Logout 登出设备。

步骤11 SDK 功能使用完后，调用 CLIENT_Cleanup 释放 SDK 资源。

注意事项

- 新增门禁卡的时候，需要保证 CardNo 和 UserID 与之前保存在设备的门禁卡信息不重复，否则会添加失败。
- 删除门禁卡的时候，如果该门禁卡绑定了人脸图片信息，需要先删除人脸图片，再删除门禁卡信息。
- 在查询门禁卡信息过程中，首先，设备必须支持该功能，且该设备本身有门禁卡信息数据。

8.2.4 示例代码

8.2.4.1 门禁卡信息的查询

```
NET_OUT_FIND_RECORD_PARAM stuOutParam= sizeof(NET_OUT_FIND_RECORD_PARAM);

NET_IN_FIND_RECORD_PARAM stuInParam= sizeof(stuInParam);



stuInParam.emType = NET_RECORD_ACCESSCTLCARD;

FIND_RECORD_ACCESSCTLCARD_CONDITION *pStuCardInfo = NEW
FIND_RECORD_ACCESSCTLCARD_CONDITION;

pStuCardInfo->dwSize = sizeof(FIND_RECORD_ACCESSCTLCARD_CONDITION);
```

```
stuInParam.pQueryCondition = (void*)pStuCardInfo;


// 开始查询，设置查询条件
bRet = CLIENT_FindRecord(m_lLoginID, &stuInParam, &stuOutParam, DEFAULT_WAIT_TIME);
if (bRet == FALSE)
{
    printf("CLIENT_FindRecord fail \n");
}


if (pStuCardInfo)
{
    delete pStuCardInfo;
    pStuCardInfo = NULL;
}


// 查询 100 条数据
NET_IN_FIND_NEXT_RECORD_PARAM stuInParam= sizeof(stuInParam);
NET_OUT_FIND_NEXT_RECORD_PARAM  stuOutParam = sizeof(stuOutParam);
stuInParam.lFindeHandle = m_lFindHandle;
stuInParam.nFileCount = 100;
memset(pAccessCardInfo, 0, stuInParam.nFileCount *
sizeof(NET_RECORDSET_ACCESS_CTL_CARD));


for (int i = 0; i < stuInParam.nFileCount; i++)
{
    pAccessCardInfo[i].dwSize = sizeof(NET_RECORDSET_ACCESS_CTL_CARD);
}
stuOutParam.pRecordList = (void*)pAccessCardInfo;
stuOutParam.nMaxRecordNum = stuInParam.nFileCount;
```

```
int nRet = CLIENT_FindNextRecord(&stuInParam, &stuOutParam, DEFAULT_WAIT_TIME);

if (!nRet)
{
printf("CLIENT_ FindNextRecord failed \n");
}
// 结束查询
CLIENT_FindRecordClose(m_lFindHandle);
```

8.2.4.2 门禁卡信息的增加、修改、删除、清除

```
// 门禁卡信息的增加
NET_CTRL_RECORDSET_INSERT_PARAM stuInParam = sizeof(stuInParam);
stuInParam.stuCtrlRecordSetInfo.dwSize = sizeof(NET_CTRL_RECORDSET_INSERT_IN);
stuInParam.stuCtrlRecordSetResult.dwSize = sizeof(NET_CTRL_RECORDSET_INSERT_OUT);
stuInParam.stuCtrlRecordSetInfo.emType = NET_RECORD_ACCESSCTLCARD;

NET_RECORDSET_ACCESS_CTL_CARD *pStrCardInfo = NEW NET_RECORDSET_ACCESS_CTL_CARD;

pStrCardInfo->dwSize = sizeof(NET_RECORDSET_ACCESS_CTL_CARD);
strncpy(pStrCardInfo->szCardNo, m_StuAddCardInfo.szCardNo, DH_MAX_CARDNO_LEN - 1);
strncpy(pStrCardInfo->szCardName, m_StuAddCardInfo.szCardName, DH_MAX_CARDNAME_LEN
- 1);
strncpy(pStrCardInfo->szUserID, m_StuAddCardInfo.szUserID, DH_MAX_USERID_LEN - 1);
strncpy(pStrCardInfo->szPsw, m_StuAddCardInfo.szPsw, DH_MAX_CARDPWD_LEN - 1);
pStrCardInfo->emStatus = NET_ACCESSCTLCARD_STATE_NORMAL;
pStrCardInfo->emType = NET_ACCESSCTLCARD_TYPE_GENERAL;
pStrCardInfo->nUserTime = m_StuAddCardInfo.nUserTime;
pStrCardInfo->bFirstEnter = TRUE;
```

```
pStrCardInfo->bIsValid = TRUE
pStrCardInfo->stuValidStartTime = m_StuAddCardInfo.stuValidStartTime;
pStrCardInfo->stuValidEndTime = m_StuAddCardInfo.stuValidEndTime;

// DoorNum 固定为 2，表示闸机的两扇门
pStrCardInfo->nDoorNum = 2;
pStrCardInfo->sznDoors[0] = 0;
pStrCardInfo->sznDoors[1] = 1;
// 控制两扇门的开门有效时间，255 表示全天
pStrCardInfo->nTimeSectionNum = 2;
pStrCardInfo->sznTimeSectionNo[0] = 255;
pStrCardInfo->sznTimeSectionNo[1] = 255;

stuInParam.stuCtrlRecordSetInfo.nBufLen = sizeof(NET_RECORDSET_ACCESS_CTL_CARD);
stuInParam.stuCtrlRecordSetInfo.pBuf = (void*)pStrCardInfo;

BOOL bRet = CLIENT_ControlDevice(m_lLoginID, DH_CTRL_RECORDSET_INSERT, &stuInParam,
DEFAULT_WAIT_TIME);
if (bRet == FALSE)
{
    Printf("CLIENT_ControlDevice insert card fail \n");
     return;
}
// 门禁卡信息的修改
NET_CTRL_RECORDSET_PARAM stuInParam = sizeof(stuInParam);
stuInParam.emType = NET_RECORD_ACCESSCTLCARD;

NET_RECORDSET_ACCESS_CTL_CARD *pStrCardInfo = NEW NET_RECORDSET_ACCESS_CTL_CARD;
memset(pStrCardInfo, 0, sizeof(NET_RECORDSET_ACCESS_CTL_CARD));
```

```
pStrCardInfo->dwSize = sizeof(NET_RECORDSET_ACCESS_CTL_CARD);
strncpy(pStrCardInfo->szCardNo, m_CardInfo.szCardNo, DH_MAX_CARDNO_LEN - 1);
strncpy(pStrCardInfo->szCardName, m_CardInfo.szCardName, DH_MAX_CARDNAME_LEN - 1);
strncpy(pStrCardInfo->szUserID, m_CardInfo.szUserID, DH_MAX_USERID_LEN - 1);
strncpy(pStrCardInfo->szPsw, m_CardInfo.szPsw, DH_MAX_CARDPWD_LEN - 1);
pStrCardInfo->emStatus = NET_ACCESSCTLCARD_STATE_NORMAL;
pStrCardInfo->emType = NET_ACCESSCTLCARD_TYPE_GENERAL;
pStrCardInfo->nUserTime = m_CardInfo.nUserTime;
pStrCardInfo->bFirstEnter = TRUE;
pStrCardInfo->bIsValid =TRUE;
pStrCardInfo->stuValidStartTime = m_CardInfo.stuValidStartTime;
pStrCardInfo->stuValidEndTime = m_CardInfo.stuValidEndTime;
pStrCardInfo->nRecNo = m_CardInfo.nRecNo;

// DoorNum 固定为 2  表示闸机的两扇门
pStrCardInfo->nDoorNum = 2;
pStrCardInfo->sznDoors[0] = 0;
pStrCardInfo->sznDoors[1] = 1;
// 控制两扇门的开门有效时间  255 表示全天
pStrCardInfo->nTimeSectionNum = 2;
pStrCardInfo->sznTimeSectionNo[0] = 255;
pStrCardInfo->sznTimeSectionNo[1] = 255;

stuInParam.nBufLen = sizeof(NET_RECORDSET_ACCESS_CTL_CARD);
stuInParam.pBuf = (void*)pStrCardInfo;

BOOL bRet = CLIENT_ControlDevice(m_lLoginID, DH_CTRL_RECORDSET_UPDATEEX, &stuInParam,
DEFAULT_WAIT_TIME);
```

```
if (FALSE == bRet)
{
    printf ("CLIENT_ControlDevice    modify cardinfo fail");
    return;
}

// 门禁卡信息的删除
NET_CTRL_RECORDSET_PARAM stuInParam = sizeof(stuInParam);
stuInParam.emType = NET_RECORD_ACCESSCTLCARD;
stuInParam.pBuf = &pCardInfo->nRecNo;
stuInParam.nBufLen = sizeof(int);
BOOL bRet = CLIENT_ControlDevice(m_lLoginID, DH_CTRL_RECORDSET_REMOVE, &stuInParam,
DEFAULT_WAIT_TIME);
if (FALSE == bRet)
{
    printf(("CLIENT_ControlDevice    delete cardinfo fail\n");
    return;
}

// 门禁卡信息的清除(删除所有门禁卡信息)
NET_CTRL_RECORDSET_PARAM stuInParam = sizeof(stuInParam);
stuInParam.emType = NET_RECORD_ACCESSCTLCARD;

BOOL bRet = CLIENT_ControlDevice(m_lLoginID, DH_CTRL_RECORDSET_CLEAR, &stuInParam,
DEFAULT_WAIT_TIME);
if (FALSE == bRet)
{
    printf ("CLIENT_ControlDevice    clear cardinfo fail\n");
     return;
```

```
}
```

8.3 人脸管理

8.3.1 简介

人脸图片的增加、修改、删除、清除，一般的使用场景和人员信息配套起来使用。

8.3.2 接口总览

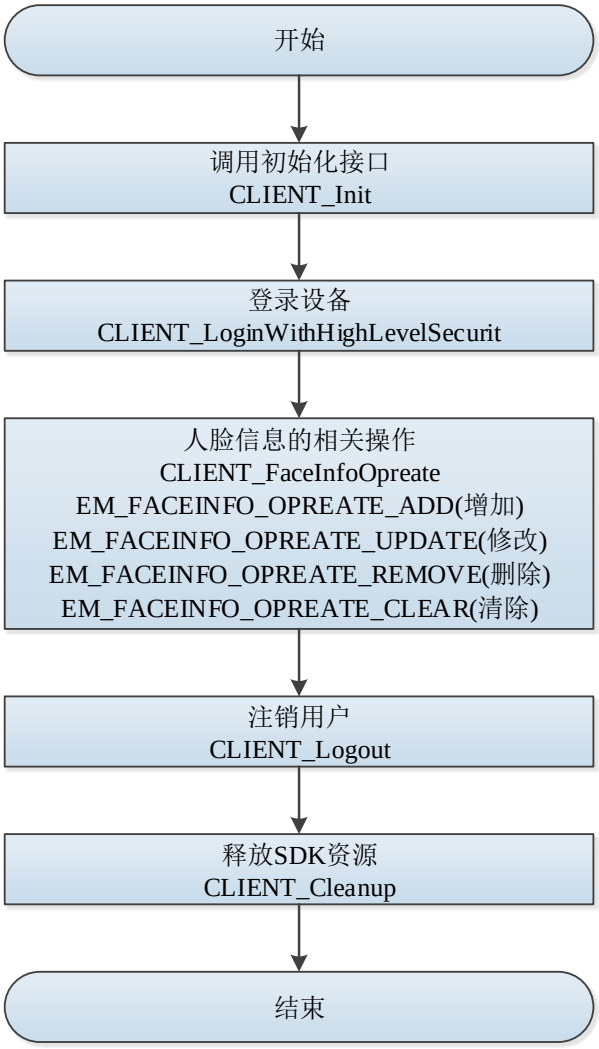
表8-2 人脸管理接口说明

接口	说明
CLIENT_FaceInfoOpreate	人脸图片信息的增加、修改、删除、清除
CLIENT_FindRecord	设置查询条件
CLIENT_FindNextRecord	在当前查询条件下查询数据
CLIENT_FindRecordClose	关闭查询
CLIENT_ControlDevice	人员信息的增加、修改、删除、清除

8.3.3 流程说明

人脸图片信息的增删改和清除流程，如图 8-2 所示。

图8-2 人脸图片信息的增删改和清除



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_FaceInfoOpreate，根据相应的枚举类型进行人脸的增加、修改、删除和清除操作。

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

- 人脸信息添加：使用的枚举为 EM_FACEINFO_OPREATE_ADD。
- 人脸信息修改：使用的枚举为 EM_FACEINFO_OPREATE_UPDATE。
- 人脸信息删除：使用的枚举为 EM_FACEINFO_OPREATE_REMOVE。

- 人脸信息清除：使用的枚举为 EM_FACEINFO_OPREATE_CLEAR。
- 人脸信息的修改和删除都是根据门禁卡的 UserID 进行操作的。

8.3.4 示例代码

```
// 人脸图片信息的增加
NET_IN_ADD_FACE_INFO stuInParam = sizeof(NET_IN_ADD_FACE_INFO);
strncpy(stuInParam.szUserID, m_StuAddCardInfo.szUserID, DH_MAX_USERID_LEN - 1);
stuInParam.stuFaceInfo.nFacePhoto = 1;

FILE *fPic = fopen(m_szFilePath, "rb");
fseek(fPic, 0, SEEK_END);
int nLength = ftell(fPic);
rewind(fPic);

stuInParam.stuFaceInfo.nFacePhotoLen[0] = nLength;
stuInParam.stuFaceInfo.pszFacePhoto[0] = new char[nLength];

memset(stuInParam.stuFaceInfo.pszFacePhoto[0], 0, nLength);
int nReadLen = fread(stuInParam.stuFaceInfo.pszFacePhoto[0], 1, nLength, fPic);
fclose(fPic);
fPic = NULL;

NET_OUT_ADD_FACE_INFO stuOutParam = sizeof(stuOutParam);

BOOL bRet = CLIENT_FaceInfoOpreate(m_lLoginID, EM_FACEINFO_OPREATE_ADD,
(void*)&stuInParam, (void*)&stuOutParam, DEFAULT_WAIT_TIME);
if (bRet == FALSE)
{
    printf("CLIENT_FaceInfoOpreate add face fail");
```

```
}

if (fPic)
{
    fclose(fPic);
    fPic = NULL;
}
if (stuInParam.stuFaceInfo.pszFacePhoto[0])
{
    delete[] stuInParam.stuFaceInfo.pszFacePhoto[0];
    stuInParam.stuFaceInfo.pszFacePhoto[0] = NULL;
}
// 人脸图片信息的修改
NET_IN_UPDATE_FACE_INFO stuInParam = sizeof(stuInParam);
strncpy(stuInParam.szUserID, m_CardInfo.szUserID, sizeof(m_CardInfo.szUserID) - 1);
stuInParam.stuFaceInfo.nFacePhoto = 1;

FILE *fPic = fopen(m_szFilePath, "rb");

fseek(fPic, 0, SEEK_END);
int nLength = ftell(fPic);
rewind(fPic);

stuInParam.stuFaceInfo.nFacePhotoLen[0] = nLength;
stuInParam.stuFaceInfo.pszFacePhoto[0] = new char[nLength];

memset(stuInParam.stuFaceInfo.pszFacePhoto[0], 0, nLength);
int nReadLen = fread(stuInParam.stuFaceInfo.pszFacePhoto[0], 1, nLength, fPic);
```

```
fclose(fPic);
fPic = NULL;

NET_OUT_UPDATE_FACE_INFO stuOutParam;
memset(&stuOutParam, 0, sizeof(stuOutParam));
stuOutParam.dwSize = sizeof(stuOutParam);

BOOL bRet = CLIENT_FaceInfoOpreate(m_lLoginID, EM_FACEINFO_OPREATE_UPDATE,
(void*)&stuInParam, (void*)&stuOutParam, DEFAULT_WAIT_TIME);
if (bRet == FALSE)
{
    printf ("CLIENT_FaceInfoOpreate modify face fail");
}

if (fPic)
{
    fclose(fPic);
    fPic = NULL;
}
if (stuInParam.stuFaceInfo.pszFacePhoto[0])
{
    delete[] stuInParam.stuFaceInfo.pszFacePhoto[0];
    stuInParam.stuFaceInfo.pszFacePhoto[0] = NULL;
}

// 人脸图片的删除
NET_IN_REMOVE_FACE_INFO stuInParam = sizeof(stuInParam);
//通过门禁卡信息的 UserID 来删除对应的人脸图片
strncpy(stuInParam.szUserID, cardInfo.szUserID, DH_MAX_USERID_LEN - 1);
```

```
NET_OUT_REMOVE_FACE_INFO stuOutParam;
memset(&stuOutParam, 0, sizeof(stuOutParam));
stuOutParam.dwSize = sizeof(stuOutParam);

bRet = CLIENT_FaceInfoOpreate(m_lLoginID, EM_FACEINFO_OPREATE_REMOVE,
(void*)&stuInParam, (void*)&stuOutParam, DEFAULT_WAIT_TIME);
if (bRet == FALSE)
{
    printf ("CLIENT_FaceInfoOpreate delete face fail");
}

// 人脸图片信息的清除(删除所有的人脸图片信息)
NET_IN_CLEAR_FACE_INFO stuInParam = sizeof(stuInParam);
NET_OUT_CLEAR_FACE_INFO stuOutParam;
memset(&stuOutParam, 0, sizeof(stuOutParam));
stuOutParam.dwSize = sizeof(stuOutParam);

BOOL bRet = CLIENT_FaceInfoOpreate(m_lLoginID, EM_FACEINFO_OPREATE_CLEAR,
(void*)&stuInParam, (void*)&stuOutParam, DEFAULT_WAIT_TIME);
if (bRet == FALSE)
{
    printf ("CLIENT_FaceInfoOpreate clear face fail");
}
```

8.4 门禁进出记录查询

8.4.1 简介

门禁记录集的查询，查询信息包括卡编号、用户序号、开门状态、卡类型、开门方式、时间等信息。

8.4.2 接口总览

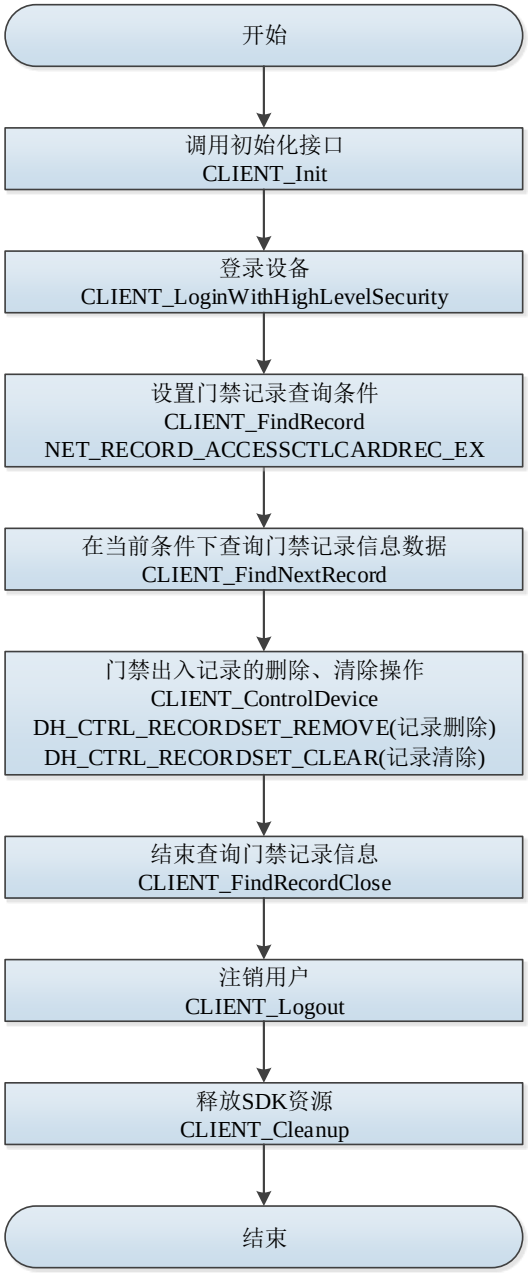
表8-3 门禁记录相关接口说明

接口	说明
CLIENT_FindRecord	设置查询条件
CLIENT_FindNextRecord	在当前查询条件下查询数据
CLIENT_FindRecordClose	关闭查询
CLIENT_ControlDevice	门禁记录的删除、清除

8.4.3 流程说明

门禁记录的查找/删除/清除流程，如图 8-3 所示。

图8-3 门禁记录的查找/删除/清除



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_FindRecord 设置门禁记录的查询条件。

步骤4　调用 CLIENT_FindNextRecord，在当前指定条件下查找固定数量的门禁记录信息。

步骤5　调用 CLIENT_ControlDevice，其中枚举类型为 DH_CTRL_RECORDSET_REMOVE（删除）和 DH_CTRL_RECORDSET_CLEAR（清除）进行门禁记录的删除和清除操作。

步骤6　查询结束后，调用 CLIENT_FindRecordClose，清除查询资源信息。

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

- 门禁记录的删除：使用的枚举为 DH_CTRL_RECORDSET_REMOVE。
- 门禁记录的清除：使用的枚举为 DH_CTRL_RECORDSET_CLEAR。

8.4.4 示例代码

查询实例代码请参见“8.2.4.1 门禁卡信息的查询”的相关示例代码。

门禁记录的删除和清除请参见“8.2.4.2 门禁卡信息的增加、修改、删除、清除”中相关删除和清除的示例代码。

第 9 章 物品监控

9.1 物品监控事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

- EVENT_IVS_OBJECT_PLACEMENT_DETECTION 物品放置检测事件
- EVENT_IVS_OBJECT_REMOVAL_DETECTION 物品拿取检测事件
- EVENT_IVS_TRASH_WITHOUT_COVER_DETECTION 垃圾桶未盖盖子检测事件

9.2 物品监控事件录像查询和下载

物品监控为智能事件的一种，故相关录像和图片查询信息请参见“2.5 查询/回放/下载录像和图片”。

调用接口 CLIENT_FindFileEx 设置查询条件。物品监控录像的查询操作、接口参数如下：

- emType：DH_FILE_QUERY_FILE 查询录像。
- pQueryCondition：NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：
  - ◇ EVENT_IVS_OBJECT_PLACEMENT_DETECTION 物品放置检测录像查询。
  - ◇ EVENT_IVS_OBJECT_REMOVAL_DETECTION 物品拿取检测录像查询。
  - ◇ EVENT_IVS_TRASH_WITHOUT_COVER_DETECTION 垃圾桶未盖盖子检测录像查询。

第 10 章 城管业务

10.1 城管业务事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

- EVENT_IVS_CITY_MOTORPARKING 城市机动车违停事件
- EVENT_IVS_CITY_NONMOTORPARKING 城市非机动车违停事件
- EVENT_IVS_DOOR_FRONT_DIRTY 门前脏乱检测事件
- EVENT_IVS_DUSTBIN_OVER_FLOW 垃圾桶满溢检测事件
- EVENT_IVS_FLOWBUSINESS 流动摊贩事件
- EVENT_IVS_GARBAGE_EXPOSURE 垃圾暴露检测事件
- EVENT_IVS_HOLD_UMBRELLA 违规撑伞检测事件
- EVENT_IVS_HUDDLE_MATERIAL 乱堆物料检测事件
- EVENT_IVS_OUTDOOR_ADVERTISEMENT 户外广告事件
- EVENT_IVS_SHOPPRESENCE 商铺占道经营事件
- EVENT_IVS_SHOP_SIGN_ABNORMAL 店招异常事件
- EVENT_IVS_SHOP_WINDOW_POST 橱窗张贴事件
- EVENT_IVS_STREET_SUNCURE 沿街晾晒事件

10.2 城管业务事件录像查询和下载

城管业务事件为智能事件的一种，故相关录像和图片查询信息请参见“2.5 查询/回放/下载录像和图片”。

调用接口 CLIENT_FindFileEx 设置查询条件。城管业务事件录像的查询操作、接口参数如下：

- emType：DH_FILE_QUERY_FILE 查询录像。
- pQueryCondition：NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：
  - ◇ EVENT_IVS_CITY_MOTORPARKING 城市机动车违停录像查询。
  - ◇ EVENT_IVS_CITY_NONMOTORPARKING 城市非机动车违停录像查询。

- ◇ EVENT_IVS_DOOR_FRONT_DIRTY 门前脏乱检测录像查询。
- ◇ EVENT_IVS_DUSTBIN_OVER_FLOW 垃圾桶满溢检测事件录像查询。
- ◇ EVENT_IVS_FLOWBUSINESS 流动摊贩录像查询。
- ◇ EVENT_IVS_GARBAGE_EXPOSURE 垃圾暴露检测录像查询。
- ◇ EVENT_IVS_HOLD_UMBRELLA 违规撑伞检测录像查询。
- ◇ EVENT_IVS_HUDDLE_MATERIAL 乱堆物料检测录像查询。
- ◇ EVENT_IVS_OUTDOOR_ADVERTISEMENT 户外广告录像查询。
- ◇ EVENT_IVS_SHOPPRESENCE 商铺占道经营录像查询。
- ◇ EVENT_IVS_SHOP_SIGN_ABNORMAL 店招异常录像查询。
- ◇ EVENT_IVS_SHOP_WINDOW_POST 橱窗张贴录像查询。
- ◇ EVENT_IVS_STREET_SUNCURE 沿街晾晒录像查询。

第 11 章 车位检测任务

11.1 车位检测事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

- EVENT_IVS_CAR_DRIVING_IN_OUT 车辆驶入驶出状态事件
- EVENT_IVS_PARKING_LOT_STATUS_DETECTION 室外停车位状态检测
- EVENT_IVS_PARKINGSPACE_STATUS 停车位状态事件
- EVENT_IVS_TRAFFIC_PARKINGSPACE_MANUALSNAP 路侧停车位手动抓图
- EVENT_IVS_TRAFFIC_PARKINGSPACENOPARKING 车位无车事件
- EVENT_IVS_TRAFFIC_PARKINGSPACEOVERLINE 车位压线事件
- EVENT_IVS_TRAFFIC_PARKINGSPACEPARKING 车位有车事件
- EVENT_IVS_TRAFFIC_ANALYSE_PRESNAP 预分析抓拍图片事件

11.2 车位检测事件录像查询和下载

车位检测为智能事件的一种，故相关录像和图片查询信息请参见“2.5 查询/回放/下载录像和图片”。

调用接口 CLIENT_FindFileEx 设置查询条件。车位检测事件录像的查询操作、接口参数如下：

- emType：DH_FILE_QUERY_FILE 查询录像。
- pQueryCondition：NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：
  - ◇ EVENT_IVS_CAR_DRIVING_IN_OUT 车辆驶入驶出状态录像查询。
  - ◇ EVENT_IVS_PARKING_LOT_STATUS_DETECTION 室外停车位状态检测录像查询。
  - ◇ EVENT_IVS_PARKINGSPACE_STATUS 停车位状态录像查询。
  - ◇ EVENT_IVS_TRAFFIC_PARKINGSPACE_MANUALSNAP 路侧停车位手动抓图录像查询。
  - ◇ EVENT_IVS_TRAFFIC_PARKINGSPACENOPARKING 车位无车录像查询。
  - ◇ EVENT_IVS_TRAFFIC_PARKINGSPACEOVERLINE 车位压线录像查询。
  - ◇ EVENT_IVS_TRAFFIC_PARKINGSPACEPARKING 车位有车录像查询。

◇ EVENT_IVS_TRAFFIC_ANALYSE_PRESNAP 预分析抓拍图片录像查询。

11.3 订阅指定的车位图片信息

11.3.1 简介

通过指定的通道号和图片 id 来订阅指定的车位图片信息或取消订阅指定的车位图片信息。

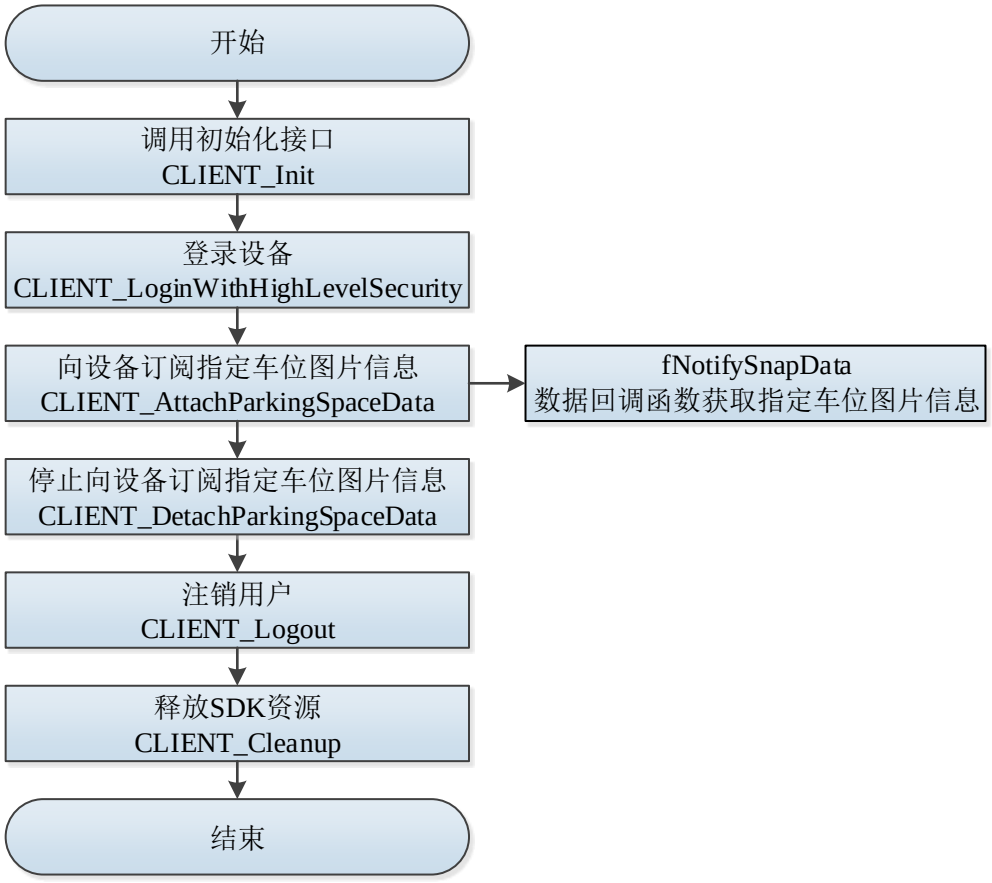
11.3.2 接口总览

表11-1 订阅指定车位图片相关接口

接口	说明
CLIENT_AttachParkingSpaceData	订阅指定的车位图片信息
CLIENT_DetachParkingSpaceData	退订指定的车位图片信息

11.3.3 流程说明

图11-1 指定车位图片信息的订阅与取消订阅



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_AttachParkingSpaceData 向设备订阅指定的车位图片信息。

步骤4　订阅成功后，设备上报的车位图片信息通过 fNotifySnapData 回调函数获取指定的车位图片并通知给用户。

步骤5　车位图片上报功能使用完毕后，调用 CLIENT_DetachParkingSpaceData 停止订阅指定车位图片信息。

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。

11.3.4 示例代码

```
void CALLBACK NotifySnapData(LLONG lParkingHandle, NET_CB_PARKINGSPACE_DATA*
pDiagnosisInfo, void* pBuf, int nBufLen, LDWORD dwUser)
{
    // 处理回调数据
}

NET_IN_ATTACH_PARKINGSPACE InParam = {sizeof (NET_IN_ATTACH_PARKINGSPACE)};
NET_OUT_ATTACH_PARKINGSPACE OutParam = {sizeof(NET_OUT_ATTACH_PARKINGSPACE)};
InParam.nChannel=0;
InParam.cbNotifySnapData = NotifySnapData;   // 订阅回调函数

// 订阅指定车位图片信息
LLONG attachHnd = CLIENT_AttachParkingSpaceData (lLoginID,&InParam,&OutParam,3000)
if(0 == attachHnd)
{
    printf("CLIENT_AttachParkingSpaceData failed! Error code %x.\n", CLIENT_GetLastError());
    return;
```

```
}


// 停止订阅热度图数据
CLIENT_DetachParkingSpaceData (attachHnd);
```

第 12 章 人群分布图

12.1 人群分布事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

EVENT_IVS_CROWDDETECTION 人群密度检测事件

12.2 人群分布事件录像查询和下载

人群分布事件为智能事件的一种，故相关录像和图片查询信息请参见“2.5 查询/回放/下载录像和图片”。

调用接口 CLIENT_FindFileEx 设置查询条件。人群分布事件录像的查询操作、接口参数如下：

- emType：DH_FILE_QUERY_FILE 查询录像。
- pQueryCondition：NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：

  EVENT_IVS_CROWDDETECTION 人群密度检测录像查询。

第 13 章 车辆密度图

13.1 车辆密度事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

- EVENT_IVS_CONGESTION_DETECTION 道路场景车辆拥堵报警事件
- EVENT_IVS_VEHICLELIMIT_DETECTION 停车场场景下停车车辆上限报警

13.2 车辆密度事件录像查询和下载

车辆密度事件为智能事件的一种，故相关录像和图片查询信息请参见“2.5 查询/回放/下载录像和图片”。

调用接口 CLIENT_FindFileEx 设置查询条件。车辆密度事件录像的查询操作、接口参数如下：

- emType：DH_FILE_QUERY_FILE 查询录像。
- pQueryCondition：NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：
  - ◇ EVENT_IVS_CONGESTION_DETECTION 道路场景车辆拥堵报警录像查询。
  - ◇ EVENT_IVS_VEHICLELIMIT_DETECTION 停车场场景下停车车辆上限录像查询。

第 14 章 热度图

14.1 订阅热度图数据

14.1.1 简介

热度图的订阅与取消。

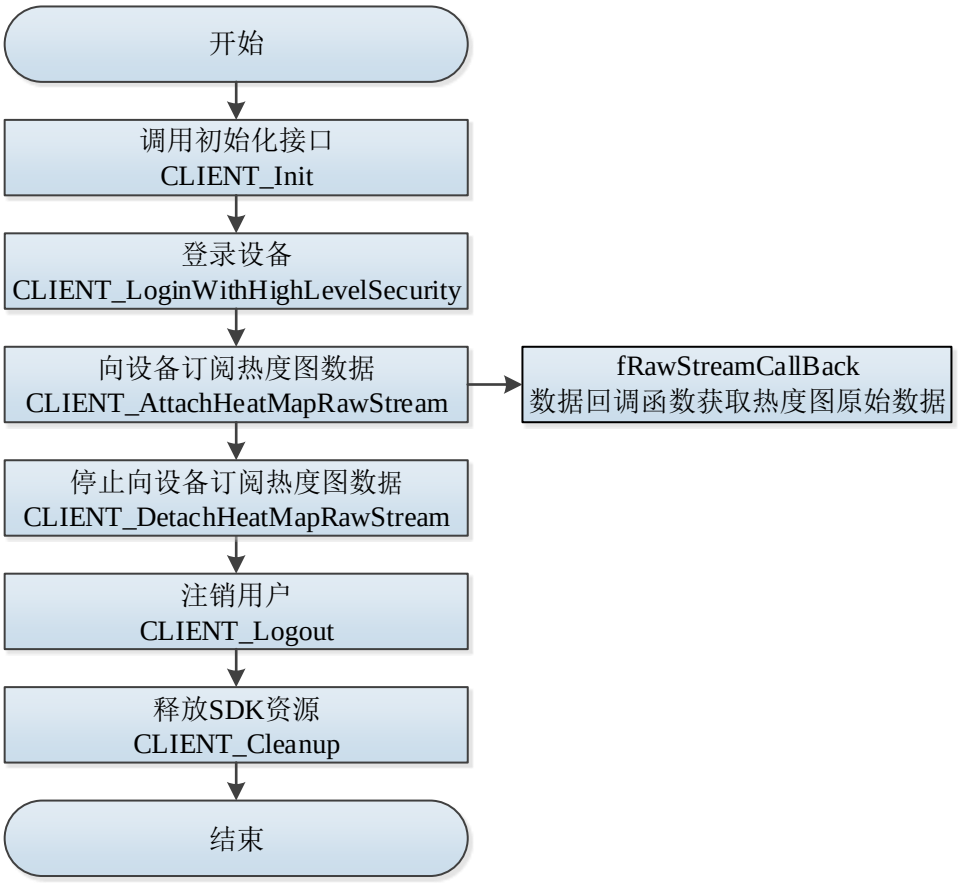
14.1.2 接口总览

表14-1 热度图相关接口说明

接口	说明
CLIENT_AttachHeatMapRawStream	订阅热度图原始数据
CLIENT_DetachHeatMapRawStream	退订热度图原始数据

14.1.3 流程说明

图14-1 热度图数据的订阅与取消订阅



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_AttachHeatMapRawStream 向设备订阅热度图数据。

步骤4　订阅成功后，设备上报的热度图数据通过 fRawStreamCallBack 回调函数获取热度图原始数据并通知给用户。

步骤5　热度图原始数据上报功能使用完毕后，调用 CLIENT_DetachHeatMapRawStream 停止订阅热度图数据。

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。

14.1.4 示例代码

```
void CALLBACK RawStreamCallBack(LLONG lAttachHandle, NET_RAWSTREAM_NOTIFY_INFO*
pBuf, DWORD dwBufLen, LDWORD dwUser)

{

    // 处理回调数据

}



NET_IN_RAWSTREAM_ATTACH_INFO InParam = {sizeof(NET_IN_RAWSTREAM_ATTACH_INFO)};

NET_OUT_RAWSTREAM_ATTACH_INFO OutParam =
{sizeof(NET_OUT_RAWSTREAM_ATTACH_INFO)};

InParam.nChannel=0;

InParam. cbNotify= RawStreamCallBack;   // 订阅回调函数



// 订阅热度图数据

LLONG attachHnd = CLIENT_AttachHeatMapRawStream(lLoginID,&InParam,&OutParam,3000)

if(0 == attachHnd)

{

    printf("CLIENT_AttachHeatMapRawStream failed! Error code %x.\n", CLIENT_GetLastError());
```

```
    return;
}

// 停止订阅热度图数据
CLIENT_DetachHeatMapRawStream(attachHnd);
```

14.2 订阅灰度图数据

14.2.1 简介

灰度图的订阅与取消。

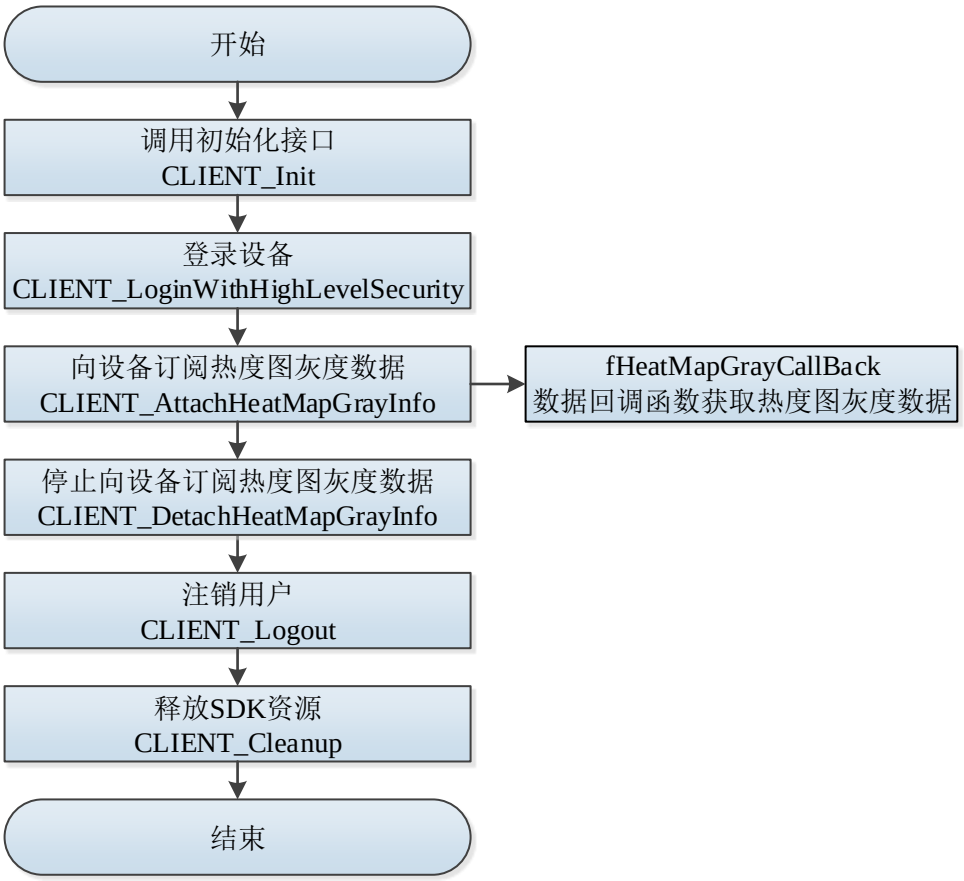
14.2.2 接口总览

表14-2 灰度图相关接口说明

接口	说明
CLIENT_AttachHeatMapGrayInfo	订阅热度图灰度数据
CLIENT_DetachHeatMapGrayInfo	退订热度图灰度数据

14.2.3 流程说明

图14-2 热度图灰度数据的订阅与取消订阅



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_AttachHeatMapGrayInfo 向设备订阅热度图灰度数据。

步骤4　订阅成功后，设备上报的热度图灰度数据通过 fHeatMapGrayCallBack 回调函数获取热度图灰度数据并通知给用户。

步骤5　热度图灰度数据上报功能使用完毕后，调用 CLIENT_DetachHeatMapGrayInfo 停止订阅热度图灰度数据。

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。

14.2.4 示例代码

```
void CALLBACK HeatMapGrayCallBack(LLONG lAttachHandle, NET_CB_HEATMAP_GRAY_INFO*
```

```
pstGrayInfo, LDWORD dwUser)
{
    // 处理回调数据
}

NET_IN_GRAY_ATTACH_INFO InParam = {sizeof (NET_IN_GRAY_ATTACH_INFO)};
NET_OUT_GRAY_ATTACH_INFO OutParam = {sizeof(NET_OUT_GRAY_ATTACH_INFO)};
InParam.nChannel=0;
InParam. cbHeatMapGray = HeatMapGrayCallBack;   // 订阅回调函数

// 订阅热度图数据
LLONG attachHnd = CLIENT_AttachHeatMapGrayInfo (lLoginID,&InParam,&OutParam,3000)
if(0 == attachHnd)
{
    printf("CLIENT_AttachHeatMapGrayInfo failed! Error code %x.\n", CLIENT_GetLastError());
    return;
}

// 停止订阅热度图数据
CLIENT_DetachHeatMapGrayInfo (attachHnd);
```

第 15 章 立体行为分析

15.1 立体行为分析事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

- EVENT_IVS_BACK_TO_DETECTION 背对检测事件
- EVENT_IVS_BIG_BAGGAGE_DETECTION 大件行李箱检测事件
- EVENT_IVS_DISTANCE_DETECTION 异常间距事件
- EVENT_IVS_PRAM_DETECTION 婴儿车检测事件
- EVENT_IVS_ABNORMALRUNDETECTION 异常奔跑事件
- EVENT_IVS_STEREO_DISTANCE_DETECTION 立体行为分析间距异常事件
- EVENT_IVS_TICKET_EVADE_DETECTION 逃票检测事件
- EVENT_IVS_WALK_DETECTION 走动检测事件
- EVENT_IVS_WRITE_ON_THE_BOARD_DETECTION 板书检测事件

15.2 立体行为分析事件录像查询和下载

立体行为分析为智能事件的一种，故相关录像和图片查询信息请参见“2.5 查询/回放/下载录像和图片”。

调用接口 CLIENT_FindFileEx 设置查询条件。立体行为分析录像的查询操作、接口参数如下：

- emType：DH_FILE_QUERY_FILE 查询录像。
- pQueryCondition：NET_IN_MEDIA_QUERY_FILE 的结构体指针，结构体中的 nEventLists 字段为：
  - ◇ EVENT_IVS_BACK_TO_DETECTION 背对检测录像查询。
  - ◇ EVENT_IVS_BIG_BAGGAGE_DETECTION 大件行李箱检测录像查询。
  - ◇ EVENT_IVS_DISTANCE_DETECTION 异常间距录像查询。
  - ◇ EVENT_IVS_PRAM_DETECTION 婴儿车检测录像查询。
  - ◇ EVENT_IVS_ABNORMALRUNDETECTION 异常奔跑录像查询。

◇ EVENT_IVS_STEREO_DISTANCE_DETECTION 立体行为分析间距异常录像查询。

◇ EVENT_IVS_TICKET_EVADE_DETECTION 逃票检测录像查询。

◇ EVENT_IVS_WALK_DETECTION 走动检测录像查询。

◇ EVENT_IVS_WRITE_ON_THE_BOARD_DETECTION 板书检测录像查询。

15.3 订阅立体行为-视频统计摘要信息

15.3.1 简介

立体行为-视频统计摘要信息事件的订阅与取消订阅。

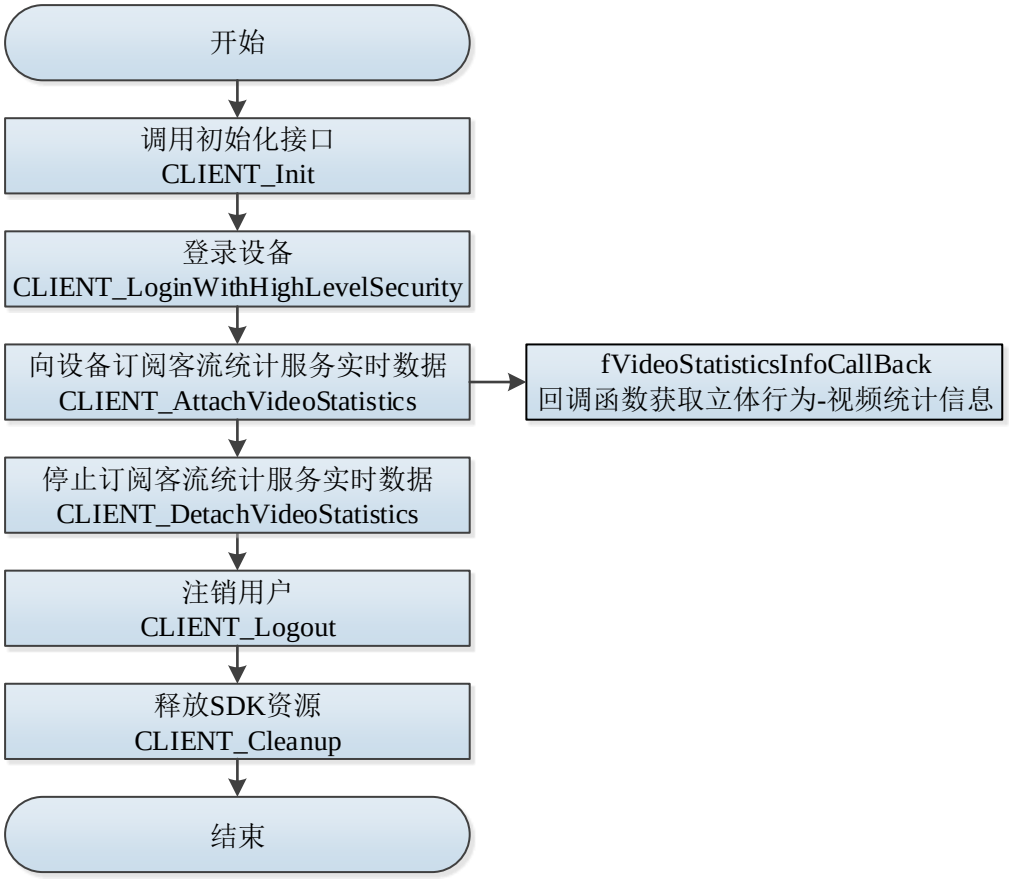
15.3.2 接口总览

表15-1 立体行为-视频统计摘要信息相关接口

接口	说明
CLIENT_AttachVideoStatistics	订阅客流统计服务实时数据
CLIENT_DetachVideoStatistics	取消订阅客流统计服务实时数据

15.3.3 流程说明

图15-1 客流统计服务实时数据的订阅与取消订阅



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_AttachVideoStatistics 向设备订阅客流统计服务实时数据。

步骤4　订阅成功后，设备上报的客流统计服务实时数据通过 fVideoStatisticsInfoCallBack 回调函数获取客流统计服务实时数据并通知给用户。

步骤5　客流统计服务实时数据上报功能使用完毕后，调用 CLIENT_DetachVideoStatistics 停止订阅客流统计服务实时数据。

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。

15.3.4 示例代码

```
void CALLBACK VideoStatisticsInfoCallBack (LLONG lAttachHandle, NET_EM_VS_TYPE emType,
void* pBuf, DWORD dwBufLen, LDWORD dwUser)
{
    // 处理回调数据
}

NET_IN_ATTACH_VIDEO_STATISTICS InParam = {sizeof (NET_IN_ATTACH_VIDEO_STATISTICS)};
NET_OUT_ATTACH_VIDEO_STATISTICS OutParam = {sizeof(NET_OUT_ATTACH_VIDEO_STATISTICS)}
InParam.nChannel=0;
InParam. cbCallBack = VideoStatisticsInfoCallBack;   // 订阅回调函数

// 订阅客流统计服务实时数据
LLONG attachHnd = CLIENT_AttachVideoStatistics (lLoginID,&InParam,&OutParam,3000)
if(0 == attachHnd)
{
    printf("CLIENT_AttachVideoStatistics failed! Error code %x.\n", CLIENT_GetLastError());
    return;
}

// 取消订阅客流统计服务实时数据
CLIENT_DetachVideoStatistics(attachHnd);
```

第 16 章 安全帽检测

16.1 安全帽检测事件订阅

具体订阅信息请参见“2.4 智能事件订阅”，在智能事件上报的回调函数 fAnalyzerDataCallBack 中过滤出通用行为事件，如下：

EVENT_IVS_HELMET_DETECTION 安全帽检测事件

第 17 章 接口函数

17.1 SDK 初始化

17.1.1 SDK 初始化 CLIENT_Init

表17-1 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	

17.1.2 SDK 清理 CLIENT_Cleanup

表17-2 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

17.1.3 设置断线重连回调函数 CLIENT_SetAutoReconnect

表17-3 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	

17.1.4 设置网络参数 CLIENT_SetNetworkParam

表17-4 设置网络参数 CLIENT_SetNetworkParam

选项	说明
描述	设置网络环境相关参数

选项	说明	
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	
说明	可根据实际网络环境，调整参数	

17.2 设备登录

17.2.1 高安全级别登录 CLIENT_LoginWithHighLevelSecurity

表17-5 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

选项	说明		
		[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 17-6。

表17-6 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

17.2.2 用户登出设备 CLIENT_Logout

表17-7 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	

17.3 实时预览

17.3.1 打开预览 CLIENT_RealPlayEx

表17-8 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);

选项	说明	
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 17-9。

表17-9 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
DH_RType_Multiplay_12	多画面预览－12 画面
DH_RType_Multiplay_25	多画面预览－25 画面

预览类型	含义
DH_RType_Multiplay_36	多画面预览－36 画面

17.3.2 关闭预览 CLIENT_StopRealPlayEx

表17-10 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	

17.3.3 保存预览数据 CLIENT_SaveRealData

表17-11 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	

17.3.4 停止保存预览数据 CLIENT_StopSaveRealData

表17-12 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	

17.3.5 设置预览数据回调 CLIENT_SetRealDataCallBackEx

表17-13 设置预览数据回调 CLIENT_SetRealDataCallBackEx

选项	说明	
描述	设置实时预览数据回调	
函数	BOOL CLIENT_SetRealDataCallBackEx(LLONG lRealHandle, fRealDataCallBackEx 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
返回值	成功返回 TRUE，失败返回 FALSE	

选项	说明
说明	无

表17-14 dwFlag 类型及含义

dwFlag	含义
0x00000001	设备的原始数据
0x00000004	转成 YUV 格式的数据

17.4 智能事件订阅

17.4.1 智能事件订阅 CLIENT_RealLoadPictureEx

表17-15 智能事件订阅 CLIENT_RealLoadPictureEx

选项	说明	
描述	订阅智能事件接口	
函数	LLONG CLIENT_RealLoadPictureEx(LLONG lLoginID, int nChannelID, DWORD dwAlarmType, BOOL bNeedPicFile, fAnalyzerDataCallBack cbAnalyzerData, LDWORD dwUser, void* Reserved);	
参数	[in] lLoginID	登录句柄
	[in] nChannelID	通道号
	[in] dwAlarmType	报警类型

选项	说明	
	[in] bNeedPicFile	是否订阅图片文件，1-订阅图片文件，在回调函数中会返回智能图片信息，0-不订阅图片文件，在回调函数中不会返回智能图片信息（在不需要图片信息时，可减少网络流量）
	[in] cbAnalyzerData	智能数据分析回调函数
	[in] dwUser	用户参数
	[in] Reserved	保留参数
返回值	成功返回 LLONG 类型的订阅句柄，失败返回 0	
说明	接口返回失败，请用 CLIENT_GetLastError 获取错误码	

本手册使用到的智能事件请参见下表。

表17-16 智能事件类型说明

预览类型	含义
EVENT_IVS_FACEDETECT	目标检测
EVENT_IVS_FACERECOGNITION	目标识别
EVENT_IVS_TRAFFICJUNCTION	交通路口事件
EVENT_IVS_TRAFFIC_PEDESTRAIN	行人事件
EVENT_IVS_TRAFFICJAM	交通拥堵事件
EVENT_IVS_TRAFFIC_OVERSPEED	超速事件
EVENT_IVS_TRAFFIC_UNDERSPEED	欠速事件
EVENT_IVS_TRAFFIC_FLOWSTATE	车流量事件
EVENT_IVS_HUMANTRAIT	人体检测事件
EVENT_IVS_CROSSLINEDETECTION	绊线入侵事件
EVENT_IVS_CROSSREGIONDETECTION	区域入侵事件
EVENT_IVS_NUMBERSTAT	人数统计事件
EVENT_IVS_MAN_NUM_DETECTION	区域内人数统计事件
EVENT_IVS_ACCESS_CTL	门禁事件

17.4.2 智能事件的取消订阅 CLIENT_StopLoadPic

表17-17 智能事件的取消订阅 CLIENT_StopLoadPic

选项	说明	
描述	停止智能事件订阅	
函数	BOOL CLIENT_StopLoadPic(LLONG lAnalyzerHandle);	
参数	[in] lAnalyzerHandle	事件订阅句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.5 智能事相关录像图片查询与下载

17.5.1 按条件查询图片或录像 CLIENT_FindFileEx

表17-18 按条件查询图片或录像 CLIENT_FindFileEx

选项	说明	
描述	按条件查找文件	
函数	LLONG CLIENT_FindFileEx(LLONG lLoginID, EM_FILE_QUERY_TYPE emType, void* pQueryCondition, void* reserved, int waittime);	
参数	[in] lLoginID	登录句柄
	[in] emType	查询的文件类型
	[in] pQueryCondition	查询条件

选项	说明	
	[in] reserved	保留字节
	[in] waittime	等待时间
返回值	成功返回 LLONG 类型的查询句柄，失败返回 0	
说明	无	

17.5.2 获取查询到文件个数 CLIENT_GetTotalFileCount

表17-19 获取查询到文件个数 CLIENT_GetTotalFileCount

选项	说明	
描述	获取查询文件的个数	
函数	BOOL CLIENT_GetTotalFileCount(LLONG lFindHandle, int* pTotalCount, void * reserved, int waittime);	
参数	[in] lFindHandle	查询句柄
	[out] pTotalCount	查询到的数目
	[in] reserved	保留字节
	[in] waittime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.5.3 查找文件 CLIENT_FindNextFileEx

表17-20 查找文件 CLIENT_FindNextFileEx

选项	说明
描述	查找文件

选项	说明	
函数	int CLIENT_FindNextFileEx(LLONG lFindHandle, int nFilecount, void* pMediaFileInfo, int maxlen, void* reserved, int waittime);	
参数	[in] lFindHandle	查询句柄
	[in] nFilecount	需要查询文件的数目
	[out] pMediaFileInfo	文件缓冲区，为查找长度的数组首地址
	[in] maxlen	查找文件数组缓冲区大小
	[in] reserved	保留字节
	[in] waittime	超时时间
返回值	成功返回查找的文件个数，失败返回-1，返回 0 表示查找结束	
说明	无	

17.5.4 结束文件查找 CLIENT_FindCloseEx

表17-21 结束文件查找 CLIENT_FindCloseEx

选项	说明	
描述	结束文件查找	
函数	BOOL CLIENT_FindCloseEx(LLONG lFindHandle);	
参数	[in] lFindHandle	查询句柄
返回值	成功返回 TRUE，失败返回 FALSE	

选项	说明
说明	无

17.5.5 开始录像回放 CLIENT_PlayBackByTimeEx2

表17-22 开始录像回放 CLIENT_PlayBackByTimeEx2

选项	说明	
描述	开始录像回放	
函数	LLONG CLIENT_PlayBackByTimeEx2(LLONG lLoginID, Int nChannelID, NET_IN_PLAY_BACK_BY_TIME_INFO* pstNetIn, NET_OUT_PLAY_BACK_BY_TIME_INFO* pstNetOut);	
参数	[in] lLoginID	登录句柄
	[in] nChannelID	通道号
	[in] pstNetIn	回放输入参数
	[out] pstNetOut	回放输出参数
返回值	成功返回 LLONG 类型的回放句柄，失败返回 0	
说明	无	

17.5.6 结束录像回放 CLIENT_StopPlayBack

表17-23 结束录像回放 CLIENT_StopPlayBack

选项	说明
描述	结束录像回放
函数	BOOL CLIENT_StopPlayBack(LLONG lPlayHandle);

选项	说明	
参数	[in] lPlayHandle	回放句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.5.7 开始录像下载 CLIENT_DownloadByTimeEx

表17-24 开始录像下载 CLIENT_DownloadByTimeEx

选项	说明	
描述	开始录像下载	
函数	LLONG CLIENT_DownloadByTimeEx(LLONG lLoginID, int nChannelId, int nRecordFileType, LPNET_TIME tmStart, LPNET_TIME tmEnd, char* sSavedFileName, fTimeDownLoadPosCallBack cbTimeDownLoadPos, LDWORD dwUserData, fDataCallBack fDownLoadDataCallBack, LDWORD dwDataUser, void* pReserved = NULL)	
参数	[in] lLoginID	登录句柄
	[in] nChannelId	通道号
	[in] nRecordFileType	录像文件类型
	[in] tmStart	录像下载开始时间
	[in] tmEnd	录像下载结束时间
	[in] sSavedFileName	指定录像数据保存路径，为空则不保存

选项	说明	
	[int] cbTimeDownLoadPos	录像下载进度回调函数
	[in] dwUserData	录像下载进度回调函数用户数据
	[in] fDownLoadDataCallBack	录像下载数据回调函数
	[in] dwDataUser	录像下载数据回调函数用户数据
	[in] pReserved	保留参数
返回值	成功返回 LLONG 类型的回放句柄，失败返回 0	
说明	无	

17.5.8 停止录像下载 CLIENT_StopDownload

表17-25 停止录像下载 CLIENT_StopDownload

选项	说明	
描述	停止录像下载	
函数	BOOL CLIENT_StopDownload(LLONG lFileHandle);	
参数	[in] lFileHandle	下载句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.5.9 下载图片 CLIENT_DownloadRemoteFile

表17-26 下载图片 CLIENT_DownloadRemoteFile

选项	说明
描述	通过文件名下载文件

选项	说明	
函数	BOOL CLIENT_DownloadRemoteFile(LLONG lLoginID, const DH_IN_DOWNLOAD_REMOTE_FILE* pInParam, DH_OUT_DOWNLOAD_REMOTE_FILE* pOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	下载文件输入参数
	[out] pOutParam	下载文件输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.6 人脸事件订阅

人脸事件订阅，请参见“17.4 智能事件订阅”。

17.7 人脸库的增删改查

17.7.1 人脸库的增删改操作 CLIENT_OperateFaceRecognitionGroup

表17-27 人脸库的增删改操作 CLIENT_OperateFaceRecognitionGroup

选项	说明
描述	人脸库的增删改操作

选项	说明	
函数	BOOL CLIENT_OperateFaceRecognitionGroup(LLONG lLoginID, const NET_IN_OPERATE_FACERECONGNITION_GROUP* pstInParam, NET_OUT_OPERATE_FACERECONGNITION_GROUP *pstOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.7.2 人脸库信息查询 CLIENT_FindGroupInfo

表17-28 人脸库信息查询 CLIENT_FindGroupInfo

选项	说明
描述	人脸库的查询操作
函数	BOOL CLIENT_FindGroupInfo(LLONG lLoginID, const NET_IN_FIND_GROUP_INFO* pstInParam, NET_OUT_FIND_GROUP_INFO *pstOutParam, int nWaitTime);

选项	说明	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8 人脸的增删改查

17.8.1 人脸的增删改操作 CLIENT_OperateFaceRecognitionDB

表17-29 人脸的增删改操作 CLIENT_OperateFaceRecognitionDB

选项	说明	
描述	人脸的增删改操作	
函数	BOOL CLIENT_OperateFaceRecognitionDB(LLONG lLoginID, const NET_IN_OPERATE_FACERECONGNITIONDB* pstInParam, NET_OUT_OPERATE_FACERECONGNITIONDB *pstOutParam, Int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8.2 设置人脸的查询条件 CLIENT_StartFindFaceRecognition

表17-30 设置人脸的查询条件 CLIENT_StartFindFaceRecognition

选项	说明	
描述	设置人脸的查询条件	
函数	BOOL CLIENT_StartFindFaceRecognition(LLONG lLoginID, const NET_IN_STARTFIND_FACERECONGNITION* pstInParam, NET_OUT_STARTFIND_FACERECONGNITION *pstOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8.3 查询人脸 CLIENT_DoFindFaceRecognition

表17-31 查询人脸 CLIENT_DoFindFaceRecognition

选项	说明
描述	查询人脸
函数	BOOL CLIENT_DoFindFaceRecognitionRecord(const NET_IN_DOFIND_FACERECONGNITIONRECORD* pstInParam, NET_OUT_DOFIND_FACERECONGNITIONRECORD *pstOutParam, int nWaitTime);

选项	说明	
参数	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8.4 结束人脸查找 CLIENT_StopFindFaceRecognition

表17-32 结束人脸查找 CLIENT_StopFindFaceRecognition

选项	说明	
描述	结束人脸查找	
函数	BOOL CLIENT_StopFindFaceRecognition(LLONG lFindHandle);	
参数	[in] lFindHandle	查询句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8.5 以库为对象进行布控 CLIENT_FaceRecognitionPutDisposition

表17-33 以库为对象进行布控 CLIENT_FaceRecognitionPutDisposition

选项	说明
描述	以库为对象进行布控

选项	说明	
函数	BOOL CLIENT_FaceRecognitionPutDisposition(LLONG lLoginID, const NET_IN_FACE_RECOGNITION_PUT_DISPOSITION_INFO* pstInParam, NET_OUT_FACE_RECOGNITION_PUT_DISPOSITION_INFO *pstOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8.6 以库为对象进行撤控 CLIENT_FaceRecognitionDelDisposition

表17-34 以库为对象进行撤控 CLIENT_FaceRecognitionDelDisposition

选项	说明
描述	以库为对象进行撤控

选项	说明	
函数	BOOL CLIENT_FaceRecognitionDelDisposition(LLONG lLoginID, const NET_IN_FACE_RECOGNITION_DEL_DISPOSITION_INFO* pstInParam, NET_OUT_FACE_RECOGNITION_DEL_DISPOSITION_INFO *pstOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8.7 以通道为对象进行布控 CLIENT_SetGroupInfoForChannel

表17-35 以通道为对象进行布控 CLIENT_SetGroupInfoForChannel

选项	说明	
描述	以通道为对象进行布控	
函数	BOOL CLIENT_SetGroupInfoForChannel(LLONG lLoginID, const NET_IN_SET_GROUPINFO_FOR_CHANNEL* pstInParam, NET_OUT_SET_GROUPINFO_FOR_CHANNEL *pstOutParam, int WaitTime);	
参数	[in] lLoginID	登录句柄

选项	说明	
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.8.8 订阅人脸查询进度 CLIENT_AttachFaceFindState

表17-36 订阅人脸查询进度 CLIENT_AttachFaceFindState

选项	说明	
描述	订阅人脸查询进度	
函数	LLONG CLIENT_AttachFaceFindState(LLONG lLoginID, const NET_IN_FACE_FIND_STATE* pstInParam, NET_OUT_FACE_FIND_STATE *pstOutParam, Int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pstOutParam	输出参数
	[in] nWaitTime	超时时间
返回值	成功返回查询进度句柄，失败返回 0	
说明	无	

17.8.9 取消订阅人脸查询进度 CLIENT_DetachFaceFindState

表17-37 取消订阅人脸查询进度 CLIENT_DetachFaceFindState

选项	说明	
描述	取消订阅人脸查询进度	
函数	BOOL CLIENT_DetachFaceFindState(LLONG lAttachHandle);	
参数	[in] lAttachHandle	CLIENT_AttachFaceFindState 返回的句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.9 人体检测

17.9.1 人体图片下载 CLIENT_DownloadRemoteFile

表17-38 人体图片下载 CLIENT_DownloadRemoteFile

选项	说明	
描述	人体图片下载	
函数	BOOL CLIENT_DownloadRemoteFile(LLONG lLoginID, const DH_IN_DOWNLOAD_REMOTE_FILE* pInParam, DH_OUT_DOWNLOAD_REMOTE_FILE* pOutParam, int nWaitTime = 1000);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入参数
	[out] pOutParam	输出参数
	[in] nWaitTime	超时时间

选项	说明
返回值	成功返回 TRUE，失败返回 FALSE
说明	无

17.10 客流量统计

17.10.1 客流量事件订阅 CLIENT_AttachVideoStatSummary

表17-39 客流量事件订阅 CLIENT_AttachVideoStatSummary

选项	说明	
描述	客流量事件订阅	
函数	LLONG CLIENT_AttachVideoStatSummary(LLONG lLoginID, const NET_IN_ATTACH_VIDEOSTAT_SUM* pInParam, NET_OUT_ATTACH_VIDEOSTAT_SUM* pOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	订阅客流输入参数
	[out] pOutParam	订阅客流输出参数
	[in] nWaitTime	超时时间
返回值	客流订阅句柄	
说明	无	

17.10.2 取消订阅客流量事件 CLIENT_DetachVideoStatSummary

表17-40 取消订阅客流量事件 CLIENT_DetachVideoStatSummary

选项	说明
描述	取消订阅客流量事件

选项	说明	
函数	BOOL CLIENT_DetachVideoStatSummary(LLONG lAttachHandle);	
参数	[in] lAttachHandle	客流订阅句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.10.3 开始查询客流历史数据（设置查询条件）

CLIENT_StartFindNumberStat

表17-41 开始查询客流历史数据（设置查询条件）CLIENT_StartFindNumberStat

选项	说明	
描述	开始查询客流历史数据（设置查询条件）	
函数	LLONG CLIENT_StartFindNumberStat(LLONG lLoginID, NET_IN_FINDNUMBERSTAT* pstInParam, NET_OUT_FINDNUMBERSTAT* pstOutParam);	
参数	[in] lLoginID	登录句柄
	[in] pstInParam	输入查询条件
	[out] pstOutParam	输出查询结果
返回值	查询句柄	
说明	无	

17.10.4 查询客流历史数据 CLIENT_DoFindNumberStat

表17-42 查询客流历史数据 CLIENT_DoFindNumberStat

选项	说明	
描述	开始查询客流历史数据（设置查询条件）	
函数	int CLIENT_DoFindNumberStat(LLONG lFindHandle, NET_IN_DOFINDNUMBERSTAT* pstInParam, NET_OUT_DOFINDNUMBERSTAT* pstOutParam);	
参数	[in] lLoginID	登录句柄
	[in] pstInParam	查询输入参数
	[out] pstOutParam	查询输出参数
返回值	查询数量	
说明	无	

17.10.5 结束查询历史数据 CLIENT_StopFindNumberStat

表17-43 结束查询历史数据 CLIENT_StopFindNumberStat

选项	说明	
描述	结束查询历史数据	
函数	BOOL CLIENT_StopFindNumberStat(LLONG lFindHandle);	
参数	[in] lFindHandle	查询句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.11 智能交通

17.11.1 开始查找数据（设置查询条件）CLIENT_FindRecord

表17-44 开始查找数据（设置查询条件）CLIENT_FindRecord

选项	说明	
描述	开始查找数据（设置查询条件）	
函数	BOOL CLIENT_FindRecord(LLONG lLoginID, NET_IN_FIND_RECORD_PARAM* pInParam, NET_OUT_FIND_RECORD_PARAM* pOutParam, int waittime=1000);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	输入查询条件
	[out] pOutParam	输出查询结果
返回值	查询句柄	
说明	无	

17.11.2 查询数据总数 CLIENT_QueryRecordCount

表17-45 查询数据总数 CLIENT_QueryRecordCount

选项	说明
描述	查询数据总数
函数	BOOL CLIENT_QueryRecordCount(NET_IN_QUEYT_RECORD_COUNT_PARAM* pInParam, NET_OUT_QUEYT_RECORD_COUNT_PARAM* pOutParam, int waittime=1000);

选项	说明	
参数	[in] pInParam	查询输入参数
	[out] pOutParam	查询输出参数
	[in] waittime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.11.3 查询指定条数数据 CLIENT_FindNextRecord

表17-46 查询指定条数数据 CLIENT_FindNextRecord

选项	说明	
描述	查询指定条数数据	
函数	int CLIENT_FindNextRecord(NET_IN_FIND_NEXT_RECORD_PARAM* pInParam, NET_OUT_FIND_NEXT_RECORD_PARAM* pOutParam, int waittime=1000);	
参数	[in] pstInParam	查询输入参数
	[out] pstOutParam	查询输出参数
	[in] waittime	超时时间
返回值	查询数量	
说明	无	

17.11.4 结束车流量查询 CLIENT_FindRecordClose

表17-47 结束车流量查询 CLIENT_FindRecordClose

选项	说明
描述	结束车流量查询

选项	说明	
函数	BOOL CLIENT_FindRecordClose(LLONG lFindHandle);	
参数	[in] lFindHandle	查询句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.11.5 黑白名单的增删改 CLIENT_OperateTrafficList

表17-48 黑白名单的增删改 CLIENT_OperateTrafficList

选项	说明	
描述	黑白名单的增删改	
函数	BOOL CLIENT_OperateTrafficList(LLONG lLoginID , NET_IN_OPERATE_TRAFFIC_LIST_RECORD* pstInParam , NET_OUT_OPERATE_TRAFFIC_LIST_RECORD *pstOutParam , int waittime)	
参数	[in] lLoginID	登录句柄
	[in] pstInParam	黑白名单操作输入参数
	[out] pstOutParam	黑白名单操作输出参数
	[in] waittime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	NET_TRAFFIC_LIST_INSERT// 增加记录操作 NET_TRAFFIC_LIST_UPDATE// 更新记录操作 NET_TRAFFIC_LIST_REMOVE// 删除记录操作	

17.11.6 批量下载文件 CLIENT_DownLoadMultiFile

表17-49 批量下载文件 CLIENT_DownLoadMultiFile

选项	说明	
描述	批量下载文件	
函数	BOOL CLIENT_DownLoadMultiFile(LLONG lLoginID, NET_IN_DOWNLOAD_MULTI_FILE *pstInParam, NET_OUT_DOWNLOAD_MULTI_FILE *pstOutParam, int waittime=1000);	
参数	[in] lLoginID	登录句柄
	[in] pstInParam	查询输入参数
	[out] pstOutParam	查询输出参数
	[in] waittime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.11.7 停止批量下载文件 CLIENT_StopLoadMultiFile

表17-50 停止批量下载文件 CLIENT_StopLoadMultiFile

选项	说明	
描述	停止批量下载文件	
函数	BOOL CLIENT_StopLoadMultiFile(LLONG lDownLoadHandle);	
参数	[in] lDownLoadHandle	批量下载句柄
返回值	成功返回 TRUE，失败返回 FALSE	

选项	说明
说明	无

17.12 门禁

17.12.1 开始查找数据（设置查询条件）CLIENT_FindRecord

表17-51 开始查找数据（设置查询条件）CLIENT_FindRecord

选项	说明	
描述	开始查找数据（设置查询条件）	
函数	BOOL CLIENT_FindRecord(LLONG lLoginID, NET_IN_FIND_RECORD_PARAM* pInParam, NET_OUT_FIND_RECORD_PARAM* pOutParam, int waittime=1000);	
参数	[in] lLoginID	登录句柄
	[int] pInParam	输入查询条件
	[out] pOutParam	输出查询结果
返回值	查询句柄	
说明	无	

17.12.2 查询指定条数数据 CLIENT_FindNextRecord

表17-52 查询指定条数数据 CLIENT_FindNextRecord

选项	说明
描述	查询指定条数数据

选项	说明	
函数	int CLIENT_FindNextRecord(NET_IN_FIND_NEXT_RECORD_PARAM* pInParam, NET_OUT_FIND_NEXT_RECORD_PARAM* pOutParam, int waittime=1000);	
参数	[int] pstInParam	查询输入参数
	[out] pstOutParam	查询输出参数
	[in] waittime	超时时间
返回值	查询数量	
说明	无	

17.12.3 结束查询 CLIENT_FindRecordClose

表17-53 结束查询 CLIENT_FindRecordClose

选项	说明	
描述	结束查询	
函数	BOOL CLIENT_FindRecordClose(LLONG lFindHandle);	
参数	[in] lFindHandle	查询句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.12.4 人员/门禁记录信息操作 CLIENT_ControlDevice

表17-54 人员/门禁记录信息操作 CLIENT_ControlDevice

选项	说明
描述	人员信息的增删改查和清除，以及门禁记录的删除和清除

选项	说明	
函数	BOOL CLIENT_ControlDevice(LLONG lLoginID , CtrlType type , void *param , int waittime = 1000);	
参数	[in] lLoginID	登录句柄
	[in] type	控制类型
	[in] param	控制参数，根据 type 不同而不同
	[in] waittime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.12.5 人脸图片的操作 CLIENT_FaceInfoOpreate

表17-55 人脸图片的操作 CLIENT_FaceInfoOpreate

选项	说明	
描述	人脸图片信息的增加、修改、删除和清除	
函数	BOOL CLIENT_FaceInfoOpreate(LLONG lLoginID, EM_FACEINFO_OPREATE_TYPE emType, void* pInParam, void* pOutParam, int nWaitTime = 1000);	
参数	[in] lLoginID	登录句柄
	[in] emType	控制类型

选项	说明	
	[in] pInParam	输入参数，根据 emType 的不同而选择不同的结构体
	[out] pOutParam	返回参数，根据 emType 的不同而选择不同的结构体
	[in] waittime	超时时间
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.13 车位检测任务

17.13.1 订阅指定的车位图片信息

表17-56 订阅指定的车位图片信息

选项	说明	
描述	订阅指定的车位图片信息	
函数	LLONG CLIENT_AttachParkingSpaceData(LLONG lLoginID, NET_IN_ATTACH_PARKINGSPACE* pstInParam, NET_OUT_ATTACH_PARKINGSPACE* pstOutParam,);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	订阅输入参数
	[out] pOutParam	订阅返回参数
返回值	车位图片订阅句柄	
说明	无	

17.13.2 取消订阅指定的车位图片信息

表17-57 取消订阅指定的车位图片信息

选项	说明	
描述	取消订阅指定的车位图片信息	
函数	BOOL CLIENT_DetachParkingSpaceData(NET_IN_DETACH_PARKINGSPACE* pstInParam， NET_OUT_DETACH_PARKINGSPACE* pstOutParam，);	
参数	[in] pstInParam	取消订阅指定车位图片输入参数
	[out] pstOutParam	取消订阅指定车位图片输出参数
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.14 热度图

17.14.1 订阅热度图数据

表17-58 订阅热度图数据

选项	说明	
描述	订阅热度图数据	
函数	LLONG CLIENT_AttachHeatMapRawStream (LLONG lLoginID, const NET_IN_RAWSTREAM_ATTACH_INFO *pInParam, NET_OUT_RAWSTREAM_ATTACH_INFO *pOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pInParam	订阅热度图输入参数

选项	说明	
	[in] nWaitTime	超时时间
	[out] pOutParam	订阅热度图返回参数
返回值	热度图订阅句柄	
说明	无	

17.14.2 取消订阅热度图数据

表17-59 取消订阅热度图数据

选项	说明	
描述	取消订阅热度图数据	
函数	BOOL CLIENT_DetachHeatMapRawStream (LLONG lAttachHandle，);	
参数	[in] lAttachHandle	订阅句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.14.3 订阅灰度图数据

表17-60 订阅灰度图数据

选项	说明
描述	订阅热度图灰度数据
函数	LLONG CLIENT_AttachHeatMapGrayInfo(LLONG lLoginID, const NET_IN_GRAY_ATTACH_INFO *pInParam, NET_OUT_GRAY_ATTACH_INFO *pOutParam, int nWaitTime);

选项	说明	
参数	[in] lLoginID	登录句柄
	[in] pInParam	订阅热度图灰度数据输入参数
	[in] nWaitTime	超时时间
	[out] pOutParam	订阅热度图灰度数据返回参数
返回值	热度图灰度数据订阅句柄	
说明	无	

17.14.4 取消订阅灰度图数据

表17-61 取消订阅灰度图数据

选项	说明	
描述	取消订阅热度图灰度数据	
函数	BOOL CLIENT_DetachHeatMapGrayInfo(LLONG lAttachHandle，);	
参数	[in] lAttachHandle	订阅句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

17.15 立体行为分析

17.15.1 订阅立体行为-视频统计摘要信息

表17-62 订阅立体行为-视频统计摘要信息

选项	说明
描述	订阅立体行为-视频统计摘要信息

选项	说明	
函数	LLONG CLIENT_AttachVideoStatistics(LLONG lLoginID, const NET_IN_ATTACH_VIDEO_STATISTICS* pstInParam, NET_OUT_ATTACH_VIDEO_STATISTICS* pstOutParam, int nWaitTime);	
参数	[in] lLoginID	登录句柄
	[in] pstInParam	订阅输入参数
	[in] nWaitTime	超时时间
	[out] pstOutParam	订阅返回参数
返回值	立体行为-视频统计摘要信息订阅句柄	
说明	无	

17.15.2 取消订阅立体行为-视频统计摘要信息

表17-63 取消订阅立体行为-视频统计摘要信息

选项	说明	
描述	取消订阅立体行为-视频统计摘要信息	
函数	BOOL CLIENT_DetachVideoStatistics(LLONG lAttachHandle，);	
参数	[in] lAttachHandle	订阅句柄
返回值	成功返回 TRUE，失败返回 FALSE	
说明	无	

第 18 章 回调函数定义

18.1 断线回调函数 fDisConnect

表18-1 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_LoginWithHighLevelSecurity 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	

18.2 断线重连回调函数 fHaveReConnect

表18-2 断线重连回调函数 fHaveReConnect

选项	说明
描述	断线重连回调函数

选项	说明	
函数	typedef void (CALLBACK *fHaveReConnect)(LLONG lLoginID, char* pchDVRIP, LONG nDVRPort, LDWORD dwUser);	
参数	[out] lLoginID	CLIENT_LoginWithHighLevelSecurity 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	

18.3 实时预览数据回调函数 fRealDataCallBackEx

表18-3 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	
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

选项	说明	
	[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	

18.4 智能事件回调函数 fAnalyzerDataCallBack

表18-4 智能事件回调函数 fAnalyzerDataCallBack

选项	说明
描述	智能事件回调函数
函数	typedef int (CALLBACK *fAnalyzerDataCallBack)(LLONG lAnalyzerHandle, DWORD dwAlarmType, void* pAlarmInfo, BYTE* pBuffer, DWORD dwBufSize, LDWORD dwUser, int nSequence, void* reserved);

选项	说明	
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] nSequence	nSequence 表示上传的相同图片情况，值为 0 时表示第一次出现，值为 2 表示最后一次出现或仅出现一次，值为 1 表示此次之后还有
	[out] reserved	保留
返回值	无	
说明	无	

18.5 回放及按文件下载进度回调函数 fDownLoadPosCallBack

表18-5 回放及按文件下载进度回调函数 fDownLoadPosCallBack

选项	说明	
描述	回放及按文件下载进度回调函数	
函数	typedef void (CALLBACK *fDownLoadPosCallBack)(LLONG lPlayHandle, DWORD dwTotalSize, DWORD dwDownLoadSize, LDWORD dwUser);	
参数	[out]lPlayHandle	回放或下载接口返回值
	[out]dwTotalSize	总大小，单位：KB

选项	说明	
	[out]dwDownLoadSize	已下载的大小，单位：KB ● -1：本次回放结束 ● -2：写文件失败
	[out]dwUser	用户数据
返回值	无	
说明	无	

18.6 回放及下载数据回调函数 fDataCallBack

表18-6 回放及下载数据回调函数 fDataCallBack

选项	说明	
描述	回放及下载数据回调函数	
函数	typedef int (CALLBACK *fDataCallBack)(LLONG lRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, LDWORD dwUser);	
参数	[out]lPlayHandle	回放或下载接口返回值
	[out] dwDataType	这里为 0（原始数据）
	[out] pBuffer	数据缓冲
	[out] dwBufSize	缓冲长度，单位：字节
	[out] dwUser	用户数据
返回值	无	
说明	无	

18.7 人脸查询进度回调函数 fFaceFindState

表18-7 人脸查询进度回调函数 fFaceFindState

选项	说明	
描述	人脸查询进度回调函数	
函数	typedef void (CALLBACK *fFaceFindState)(LLONG lLoginID, LLONG lAttachHandle, NET_CB_FACE_FIND_STATE* pstStates, int nStateNum, LDWORD dwUser);	
参数	[out] lLoginID	返回登录句柄
	[out] lAttachHandle	事件订阅句柄
	[out] pstStates	人脸查询的状态信息
	[out] nStateNum	人脸查询的进度
	[out] dwUser	用户数据
返回值	无	
说明	无	

18.8 客流量事件订阅回调 fVideoStatSumCallBack

表18-8 客流量事件订阅回调 fVideoStatSumCallBack

选项	说明
描述	客流量事件订阅回调

选项	说明	
函数	typedef void (CALLBACK *fVideoStatSumCallBack) (LLONG lAttachHandle, NET_VIDEOSTAT_SUMMARY* pBuf, DWORD dwBufLen, LDWORD dwUser);	
参数	[out] lAttachHandle	客流订阅句柄
	[out] pBuf	客流返回数据
	[out] dwBufLen	返回数据长度
	[out] dwUser	用户数据
返回值	无	
说明	无	

18.9 批量下载文件进度回调函数 fMultiFileDownLoadPosCB

表18-9 批量下载文件进度回调函数 fMultiFileDownLoadPosCB

选项	说明
描述	批量下载文件进度回调函数
函数	typedef void (CALLBACK *fMultiFileDownLoadPosCB)(LLONG lDownLoadHandle, DWORD dwID, DWORD dwFileTotalSize, DWORD dwDownLoadSize, int nError, LDWORD dwUser, void* pReserved);

选项	说明	
参数	[out] lDownLoadHandle	批量下载文件句柄
	[out] dwID	ID 为用户设置的 dwFileID
	[out] dwFileTotalSize	下载文件的总大小
	[out] dwDownLoadSize	当前文件下载大小，当该值为该类型的最大值时表示下载结束
	[out] nError	下载出错：1-缓存不足，2-对返回数据的校验出错，3-下载当前文件失败，4-创建对应保存文件失败
	[out] dwUser	用户数据
	[out] pReserved	保留字段
返回值	无	
说明	无	

18.10 指定车位图片信息事件订阅回调函数

表18-10 指定车位图片信息事件订阅回调函数

选项	说明	
描述	指定车位图片信息事件订阅回调	
函数	typedef int (CALLBACK *fNotifySnapData) (LLONG lParkingHandle, NET_CB_PARKINGSPACE_DATA* pDiagnosisInfo, void* pBuf, int nBufLen， LDWORD dwUser);	
参数	[out] lParkingHandle	车位图片信息订阅句柄
	[out] pDiagnosisInfo	车位图片信息返回数据

选项	说明	
	[out] pBuf	数据缓冲
	[out] nBufLen	缓冲长度
	[out] dwUser	用户数据
返回值	无	
说明	无	

18.11 热度图原始数据事件订阅回调函数

表18-11 热度图原始数据事件订阅回调函数

选项	说明	
描述	热度图原始数据事件订阅回调	
函数	typedef void(CALLBACK *fRawStreamCallBack) (LLONG lAttachHandle, NET_RAWSTREAM_NOTIFY_INFO* pBuf, DWORD dwBufLen, LDWORD dwUser);	
参数	[out] lAttachHandle	热度图原始数据订阅句柄
	[out] pBuf	返回热度图原始数据
	[out] dwBufLen	返回数据长度
	[out] dwUser	用户数据
返回值	无	
说明	无	

18.12 热度图灰度数据事件订阅回调函数

表18-12 热度图灰度数据事件订阅回调函数

选项	说明	
描述	热度图灰度数据事件订阅回调	
函数	typedef void(CALLBACK *fHeatMapGrayCallBack) (LLONG lAttachHandle, NET_CB_HEATMAP_GRAY_INFO* pstGrayInfo, LDWORD dwUser);	
参数	[out] lAttachHandle	热度图灰度数据订阅句柄
	[out] pstGrayInfo	返回灰度数据
	[out] dwUser	用户数据
返回值	无	
说明	无	

18.13 立体行为视频统计摘要信息事件订阅回调函数

表18-13 立体行为视频统计摘要信息事件订阅回调函数

选项	说明
描述	立体行为视频统计摘要信息事件订阅回调
函数	typedef void(CALLBACK *fVideoStatisticsInfoCallBack) (LLONG lAttachHandle, NET_EM_VS_TYPE emType, void* pBuf, DWORD nBufLen， LDWORD dwUser);

选项	说明	
参数	[out] lAttachHandle	立体行为视频统计摘要信息订阅句柄
	[out] emType	业务类型
	[out] pBuf	数据缓冲
	[out] nBufLen	缓冲长度
	[out] dwUser	用户数据
返回值	无	
说明	无	

附录1 智能事件

智能事件名称	序号	含义
EVENT_IVS_ALL	0x00000001	订阅所有事件
EVENT_IVS_CROSSLINEDETECTION	0x00000002	绊线入侵事件(对应 DEV_EVENT_CROSSLINE_INFO)
EVENT_IVS_CROSSREGIONDETECTION	0x00000003	区域入侵事件(对应 DEV_EVENT_CROSSREGION_INFO)
EVENT_IVS_PASTEDETECTION	0x00000004	贴条事件(对应 DEV_EVENT_PASTE_INFO)
EVENT_IVS_LEFTDETECTION	0x00000005	物品遗留事件(对应 DEV_EVENT_LEFT_INFO)
EVENT_IVS_STAYDETECTION	0x00000006	停留事件(对应 DEV_EVENT_STAY_INFO)
EVENT_IVS_WANDERDETECTION	0x00000007	徘徊事件(对应 DEV_EVENT_WANDER_INFO)
EVENT_IVS_PRESERVATION	0x00000008	物品保全事件(对应 DEV_EVENT_PRESERVATION_INFO)
EVENT_IVS_MOVEDETECTION	0x00000009	移动事件(对应 DEV_EVENT_MOVE_INFO)
EVENT_IVS_TAILDETECTION	0x0000000A	尾随事件(对应 DEV_EVENT_TAIL_INFO)
EVENT_IVS_RIOTERDETECTION	0x0000000B	聚众事件(对应 DEV_EVENT_RIOTERL_INFO)
EVENT_IVS_FIREDETECTION	0x0000000C	火警事件(对应 DEV_EVENT_FIRE_INFO)
EVENT_IVS_SMOKEDETECTION	0x0000000D	烟雾报警事件(对应 DEV_EVENT_SMOKE_INFO)
EVENT_IVS_FIGHTDETECTION	0x0000000E	斗殴事件(对应 DEV_EVENT_FIGHT_INFO)
EVENT_IVS_FLOWSTAT	0x0000000F	流量统计事件(对应 DEV_EVENT_FLOWSTAT_INFO)
EVENT_IVS_NUMBERSTAT	0x00000010	数量统计事件(对应 DEV_EVENT_NUMBERSTAT_INFO)
EVENT_IVS_CAMERACOVERDDETECTION	0x00000011	摄像头覆盖事件(保留)

智能事件名称	序号	含义
EVENT_IVS_CAMERAMOVEDDETECTION	0x00000012	摄像头移动事件(保留)
EVENT_IVS_VIDEOABNORMALDETECTION	0x00000013	视频异常事件(对应 DEV_EVENT_VIDEOABNORMALDETECTION_INFO)
EVENT_IVS_VIDEOBADDETECTION	0x00000014	视频损坏事件(保留)
EVENT_IVS_TRAFFICCONTROL	0x00000015	交通管理事件(对应 DEV_EVENT_TRAFFICCONTROL_INFO)
EVENT_IVS_TRAFFICACCIDENT	0x00000016	交通事故事件(对应 DEV_EVENT_TRAFFICACCIDENT_INFO)
EVENT_IVS_TRAFFICJUNCTION	0x00000017	交通路口事件----老规则(对应 DEV_EVENT_TRAFFICJUNCTION_INFO)
EVENT_IVS_TRAFFICGATE	0x00000018	交通卡口事件----老规则(对应 DEV_EVENT_TRAFFICGATE_INFO)
EVENT_TRAFFICSNAPSHOT	0x00000019	交通抓拍事件(对应 DEV_EVENT_TRAFFICSNAPSHOT_INFO)
EVENT_IVS_FACEDETECT	0x0000001A	目标检测事件(对应 DEV_EVENT_FACEDETECT_INFO)
EVENT_IVS_TRAFFICJAM	0x0000001B	交通拥堵事件(对应 DEV_EVENT_TRAFFICJAM_INFO)
EVENT_IVS_TRAFFIC_NONMOTORINMOTORROUTE	0x0000001C	非机动车占机动车车道事件(对应 DEV_EVENT_TRAFFIC_NONMOTORINMOTORROUTE_INFO)
EVENT_IVS_TRAFFIC_RUNREDLIGHT	0x00000100	交通违章-闯红灯事件(对应 DEV_EVENT_TRAFFIC_RUNREDLIGHT_INFO)
EVENT_IVS_TRAFFIC_OVERLINE	0x00000101	交通违章-压车道线事件(对应 DEV_EVENT_TRAFFIC_OVERLINE_INFO)
EVENT_IVS_TRAFFIC_RETROGRADE	0x00000102	交通违章-逆行事件(对应 DEV_EVENT_TRAFFIC_RETROGRADE_INFO)
EVENT_IVS_TRAFFIC_TURNLEFT	0x00000103	交通违章-违章左转(对应 DEV_EVENT_TRAFFIC_TURNLEFT_INFO)
EVENT_IVS_TRAFFIC_TURNRIGHT	0x00000104	交通违章-违章右转(对应 DEV_EVENT_TRAFFIC_TURNRIGHT_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_UTURN	0x00000105	交通违章-违章掉头(对应 DEV_EVENT_TRAFFIC_UTURN_INFO)
EVENT_IVS_TRAFFIC_OVERSPEED	0x00000106	交通违章-超速(对应 DEV_EVENT_TRAFFIC_OVERSPEED_INFO)
EVENT_IVS_TRAFFIC_UNDERSPEED	0x00000107	交通违章-低速(对应 DEV_EVENT_TRAFFIC_UNDERSPEED_INFO)
EVENT_IVS_TRAFFIC_PARKING	0x00000108	交通违章-违章停车(对应 DEV_EVENT_TRAFFIC_PARKING_INFO)
EVENT_IVS_TRAFFIC_WRONGROUTE	0x00000109	交通违章-不按车道行驶(对应 DEV_EVENT_TRAFFIC_WRONGROUTE_INFO)
EVENT_IVS_TRAFFIC_CROSSLANE	0x0000010A	交通违章-违章变道(对应 DEV_EVENT_TRAFFIC_CROSSLANE_INFO)
EVENT_IVS_TRAFFIC_OVERYELLOWLINE	0x0000010B	交通违章-压黄线(对应 DEV_EVENT_TRAFFIC_OVERYELLOWLINE_INFO)
EVENT_IVS_TRAFFIC_DRIVINGONSHOULDER	0x0000010C	交通违章-路肩行驶事件(对应 DEV_EVENT_TRAFFIC_DRIVINGONSHOULDER_INFO)
EVENT_IVS_TRAFFIC_YELLOWPLATEINLANE	0x0000010E	交通违章-黄牌车占道事件(对应 DEV_EVENT_TRAFFIC_YELLOWPLATEINLANE_INFO)
EVENT_IVS_TRAFFIC_PEDESTRAINPRIORITY	0x0000010F	交通违章-礼让行人/斑马线行人优先事件(对应 DEV_EVENT_TRAFFIC_PEDESTRAINPRIORITY_INFO)
EVENT_IVS_ELECTROSPARKDETECTION	0x00000110	电火花事件(对应 DEV_EVENT_ELECTROSPARK_INFO)
EVENT_IVS_TRAFFIC_NOPASSING	0x00000111	交通违章-禁止通行事件(对应 DEV_EVENT_TRAFFIC_NOPASSING_INFO)
EVENT_IVS_ABNORMALRUNDETECTION	0x00000112	异常奔跑事件(对应 DEV_EVENT_ABNORMALRUNDETECTION_INFO)
EVENT_IVS_RETROGRADEDETECTION	0x00000113	人员逆行事件(对应 DEV_EVENT_RETROGRADEDETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_INREGIONDETECTION	0x00000114	区域内检测事件(对应 DEV_EVENT_INREGIONDETECTION_INFO)
EVENT_IVS_TAKENAWAYDETECTION	0x00000115	物品搬移事件(对应 DEV_EVENT_TAKENAWAYDETECTION_INFO)
EVENT_IVS_PARKINGDETECTION	0x00000116	非法停车事件(对应 DEV_EVENT_PARKINGDETECTION_INFO)
EVENT_IVS_FACERECOGNITION	0x00000117	目标识别事件(对应 DEV_EVENT_FACERECOGNITION_INFO)
EVENT_IVS_TRAFFIC_MANUALSNAP	0x00000118	交通手动抓拍事件(对应 DEV_EVENT_TRAFFIC_MANUALSNAP_INFO)
EVENT_IVS_TRAFFIC_FLOWSTATE	0x00000119	交通流量统计事件(对应 DEV_EVENT_TRAFFIC_FLOW_STATE)
EVENT_IVS_TRAFFIC_STAY	0x0000011A	交通滞留事件(对应 DEV_EVENT_TRAFFIC_STAY_INFO)
EVENT_IVS_TRAFFIC_VEHICLEINROUTE	0x0000011B	有车占道事件(对应 DEV_EVENT_TRAFFIC_VEHICLEINROUTE_INFO)
EVENT_ALARM_MOTIONDETECT	0x0000011C	视频移动侦测事件(对应 DEV_EVENT_ALARM_INFO)
EVENT_ALARM_LOCALALARM	0x0000011D	外部报警事件(对应 DEV_EVENT_ALARM_INFO)
EVENT_IVS_PSRISEDETECTION	0x0000011E	囚犯起身事件(对应 DEV_EVENT_PSRISEDETECTION_INFO)
EVENT_IVS_CROSSFENCEDETECTION	0x0000011F	翻越围栏事件(对应 DEV_EVENT_CROSSFENCEDETECTION_INFO)
EVENT_IVS_TRAFFIC_TOLLGATE	0x00000120	交通违章-卡口事件----新规则(对应 DEV_EVENT_TRAFFICJUNCTION_INFO)
EVENT_IVS_DENSITYDETECTION	0x00000121	人员密集度检测(对应 DEV_EVENT_DENSITYDETECTION_INFO)
EVENT_IVS_VIDEODIAGNOSIS	0x00000122	视频诊断结果事件(对应 NET_VIDEODIAGNOSIS_COMMON_INFO 和 NET_REAL_DIAGNOSIS_RESULT)
EVENT_IVS_QUEUEDETECTION	0x00000123	排队检测报警事件(对应 DEV_EVENT_QUEUEDETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_VEHICLEINBUSROUTE	0x00000124	占用公交车道事件(对应 DEV_EVENT_TRAFFIC_VEHICLEINBUSROUTE_INFO)
EVENT_IVS_TRAFFIC_BACKING	0x00000125	违章倒车事件(对应 DEV_EVENT_IVS_TRAFFIC_BACKING_INFO)
EVENT_IVS_AUDIO_ABNORMALDETECTION	0x00000126	声音异常检测(对应 DEV_EVENT_IVS_AUDIO_ABNORMALDETECTION_INFO)
EVENT_IVS_TRAFFIC_RUNYELLOWLIGHT	0x00000127	交通违章-闯黄灯事件(对应 DEV_EVENT_TRAFFIC_RUNYELLOWLIGHT_INFO)
EVENT_IVS_CLIMBDETECTION	0x00000128	攀高检测事件(对应 DEV_EVENT_IVS_CLIMB_INFO)
EVENT_IVS_LEAVEDETECTION	0x00000129	离岗检测事件(对应 DEV_EVENT_IVS_LEAVE_INFO)
EVENT_IVS_TRAFFIC_PARKINGONYELLOWBOX	0x0000012A	黄网格线抓拍事件(对应 DEV_EVENT_TRAFFIC_PARKINGONYELLOWBOX_INFO)
EVENT_IVS_TRAFFIC_PARKINGSPACEPARKING	0x0000012B	车位有车事件(对应 DEV_EVENT_TRAFFIC_PARKINGSPACEPARKING_INFO)
EVENT_IVS_TRAFFIC_PARKINGSPACENOPARKING	0x0000012C	车位无车事件(对应 DEV_EVENT_TRAFFIC_PARKINGSPACENOPARKING_INFO)
EVENT_IVS_TRAFFIC_PEDESTRAIN	0x0000012D	交通行人事件(对应 DEV_EVENT_TRAFFIC_PEDESTRAIN_INFO)
EVENT_IVS_TRAFFIC_THROW	0x0000012E	交通抛洒物品事件(对应 DEV_EVENT_TRAFFIC_THROW_INFO)
EVENT_IVS_TRAFFIC_IDLE	0x0000012F	交通空闲事件(对应 DEV_EVENT_TRAFFIC_IDLE_INFO)
EVENT_ALARM_VEHICLEACC	0x00000130	车载 ACC 断电报警事件(对应 DEV_EVENT_ALARM_VEHICLEACC_INFO)
EVENT_ALARM_VEHICLE_TURNOVER	0x00000131	车辆侧翻报警事件(对应 DEV_EVENT_VEHICEL_ALARM_INFO)

智能事件名称	序号	含义
EVENT_ALARM_VEHICLE_COLLISION	0x00000132	车辆撞车报警事件(对应 DEV_EVENT_VEHICEL_ALARM_INFO)
EVENT_ALARM_VEHICLE_LARGE_ANGLE	0x00000133	车载摄像头大角度扭转事件
EVENT_IVS_TRAFFIC_PARKINGSPACEOVERLINE	0x00000134	车位压线事件(对应 DEV_EVENT_TRAFFIC_PARKINGSPACEOVERLINE_INFO)
EVENT_IVS_MULTISCENESWITCH	0x00000135	多场景切换事件(对应 DEV_EVENT_IVS_MULTI_SCENE_SWICH_INFO)
EVENT_IVS_TRAFFIC_RESTRICTED_PLATE	0x00000136	受限车牌事件(对应 DEV_EVENT_TRAFFIC_RESTRICTED_PLATE)
EVENT_IVS_TRAFFIC_OVERSTOPLINE	0x00000137	压停止线事件(对应 DEV_EVENT_TRAFFIC_OVERSTOPLINE)
EVENT_IVS_TRAFFIC_WITHOUT_SAFEBELT	0x00000138	交通未系安全带事件(对应 DEV_EVENT_TRAFFIC_WITHOUT_SAFEBELT)
EVENT_IVS_TRAFFIC_DRIVER_SMOKING	0x00000139	驾驶员抽烟事件(对应 DEV_EVENT_TRAFFIC_DRIVER_SMOKING)
EVENT_IVS_TRAFFIC_DRIVER_CALLING	0x0000013A	驾驶员打电话事件(对应 DEV_EVENT_TRAFFIC_DRIVER_CALLING)
EVENT_IVS_TRAFFIC_PEDESTRAINRUNREDLIGHT	0x0000013B	行人闯红灯事件(对应 DEV_EVENT_TRAFFIC_PEDESTRAINRUNREDLIGHT_INFO)
EVENT_IVS_TRAFFIC_PASSNOTINORDER	0x0000013C	未按规定依次通行(对应 DEV_EVENT_TRAFFIC_PASSNOTINORDER_INFO)
EVENT_IVS_OBJECT_DETECTION	0x00000141	物体特征检测事件
EVENT_ALARM_ANALOGALARM	0x00000150	模拟量报警通道的报警事件(对应 DEV_EVENT_ALARM_ANALOGALRM_INFO)
EVENT_IVS_CROSSLINEDETECTION_EX	0x00000151	绊线入侵扩展事件
EVENT_ALARM_COMMON	0x00000152	普通录像
EVENT_ALARM_VIDEOBLIND	0x00000153	视频遮挡事件(对应 DEV_EVENT_ALARM_VIDEOBLIND)

智能事件名称	序号	含义
EVENT_ALARM_VIDEOLOSS	0x00000154	视频丢失事件
EVENT_IVS_GETOUTBEDDETECTION	0x00000155	下床事件(对应 DEV_EVENT_GETOUTBED_INFO)
EVENT_IVS_PATROLDETECTION	0x00000156	巡逻检测事件(对应 DEV_EVENT_PATROL_INFO)
EVENT_IVS_ONDUTYDETECTION	0x00000157	站岗检测事件(对应 DEV_EVENT_ONDUTY_INFO)
EVENT_IVS_NOANSWERCALL	0x00000158	门口机呼叫未响应事件
EVENT_IVS_STORAGENOTEXIST	0x00000159	存储组不存在事件
EVENT_IVS_STORAGELOWSPACE	0x0000015A	硬盘空间低报警事件
EVENT_IVS_STORAGEFAILURE	0x0000015B	存储错误事件
EVENT_IVS_PROFILEALARMTRANSMIT	0x0000015C	报警传输事件(对应 DEV_EVENT_PROFILE_ALARM_TRANSMIT_INFO)
EVENT_IVS_VIDEOSTATIC	0x0000015D	视频静态检测事件(对应 DEV_EVENT_ALARM_VIDEOSTATIC_INFO)
EVENT_IVS_VIDEOTIMING	0x0000015E	视频定时检测事件(对应 DEV_EVENT_ALARM_VIDEOTIMING_INFO)
EVENT_IVS_HEATMAP	0x0000015F	热度图(对应 CFG_IVS_HEATMAP_INFO)
EVENT_IVS_CITIZENIDCARD	0x00000160	证件信息读取事件(对应 DEV_EVENT_ALARM_CITIZENIDCARD_INFO)
EVENT_IVS_PICINFO	0x00000161	图片信息事件(对应 DEV_EVENT_ALARM_PIC_INFO)
EVENT_IVS_NETPLAYCHECK	0x00000162	上网登记事件(对应 DEV_EVENT_ALARM_NETPLAYCHECK_INFO)
EVENT_IVS_TRAFFIC_JAM_FORBID_INTO	0x00000163	车辆拥堵禁入事件(对应 DEV_EVENT_ALARM_JAMFORBIDINTO_INFO)
EVENT_IVS_SNAPBYTIME	0x00000164	定时抓图事件(对应 DEV_EVENT_SNAPBYTIME)

智能事件名称	序号	含义
EVENT_IVS_PTZ_PRESET	0x00000165	云台转动到预置点事件(对应 DEV_EVENT_ALARM_PTZ_PRESET_INFO)
EVENT_IVS_RFID_INFO	0x00000166	红外线检测信息事件(对应 DEV_EVENT_ALARM_RFID_INFO)
EVENT_IVS_STANDUPDETECTION	0x00000167	人起立检测事件
EVENT_IVS_QSYTRAFFICCARWEIGHT	0x00000168	交通卡口称重事件(对应 DEV_EVENT_QSYTRAFFICCARWEIGHT_INFO)
EVENT_IVS_TRAFFIC_COMPAREPLATE	0x00000169	卡口前后车牌合成事件(对应 DEV_EVENT_TRAFFIC_COMPAREPLATE_INFO)
EVENT_IVS_SHOOTINGSCORERECOGNITION	0x0000016A	打靶像机事件(对应 DEV_EVENT_SHOOTING_SCORE_RECOGNITION_INFO,CFG_IVS_SHOOTINGSCORERECOGNITION_INFO)
EVENT_IVS_TRAFFIC_FCC	0x0000016B	加油站提枪、挂枪事件(对应 DEV_EVENT_TRAFFIC_FCC_INFO)
EVENT_IVS_TRAFFIC_TRANSFINITE	0x0000016C	违章超限抓图上报事件(对应 DEV_EVENT_TRAFFIC_TRANSFINITE_INFO)
EVENT_IVS_SCENE_CHANGE	0x0000016D	场景变更事件(对应 DEV_ALRAM_SCENECHANGE_INFO,CFG_VIDEOABNORMALDETECTION_INFO)
EVENT_IVS_LETRACK	0x0000016E	简单跟踪事件(暂未有具体事件)
EVENT_IVS_OBJECT_ACTION	0x0000016F	物体检测事件(暂未有具体事件)
EVENT_IVS_TRAFFIC_ANALYSE_PRESNAP	0x00000170	预分析抓拍图片事件(对应 DEV_EVENT_TRAFFIC_ANALYSE_PRESNAP_INFO)
EVENT_ALARM_EQSTATE	0x00000171	智能插座电量状态上报(暂未有具体事件)
EVENT_IVS_ALARM_IPC	0x00000172	DVR/NVR 设备上的 IPC 报警(对应 DEV_EVENT_ALARM_IPC_INFO)
EVENT_IVS_POS_RECORD	0x00000173	POS 录像查询事件(暂未有具体事件)
EVENT_IVS_NEAR_DISTANCE_DETECTION	0x00000174	近距离接触事件(对应 DEV_EVENT_NEAR_DISTANCE_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_OBJECTSTRUCTLIZE_PERSON	0x00000175	行人特征检测事件(对应 DEV_EVENT_OBJECTSTRUCTLIZE_PERSON_INFO)
EVENT_IVS_OBJECTSTRUCTLIZE_NONMOTOR	0x00000176	非机动车特征检测事件(对应 DEV_EVENT_OBJECTSTRUCTLIZE_NONMOTOR_INFO)
EVENT_IVS_TUMBLE_DETECTION	0x00000177	倒地报警事件(对应 DEV_EVENT_TUMBLE_DETECTION_INFO)
EVENT_IVS_TRAFFIC_ALL	0x000001FF	所有以 traffic 开头的事件,目前指的是
EVENT_IVS_VIDEOANALYSE	0x00000200	所有智能分析事件
EVENT_IVS_LINKSD	0x00000201	LinkSD 事件(对应 DEV_EVENT_LINK_SD)
EVENT_IVS_VEHICLEANALYSE	0x00000202	车辆特征检测分析(对应 DEV_EVENT_VEHICLEANALYSE)
EVENT_IVS_FLOWRATE	0x00000203	流量使用情况事件(对应 DEV_EVENT_FLOWRATE_INFO)
EVENT_IVS_ACCESS_CTL	0x00000204	门禁事件(对应 DEV_EVENT_ACCESS_CTL_INFO)
EVENT_IVS_SNAPMANUAL	0x00000205	SnapManual 事件(对应 DEV_EVENT_SNAPMANUAL)
EVENT_IVS_TRAFFIC_ELETAGINFO	0x00000206	RFID 电子车牌标签事件(对应 DEV_EVENT_TRAFFIC_ELETAGINFO_INFO)
EVENT_IVS_TRAFFIC_TIREDPHYSIOLOGICAL	0x00000207	生理疲劳驾驶事件(对应 DEV_EVENT_TIREDPHYSIOLOGICAL_INFO)
EVENT_IVS_TRAFFIC_BUSSHARPTURN	0x00000208	车辆急转报警事件(对应 DEV_EVENT_BUSSHARPTURN_INFO)
EVENT_IVS_CITIZEN_PICTURE_COMPARE	0x00000209	人证比对事件(对应 DEV_EVENT_CITIZEN_PICTURE_COMPARE_INFO)
EVENT_IVS_TRAFFIC_TIREDLOWERHEAD	0x0000020A	开车低头报警事件(对应 DEV_EVENT_TIREDLOWERHEAD_INFO)
EVENT_IVS_TRAFFIC_DRIVERLOOKAROUND	0x0000020B	开车左顾右盼报警事件(对应 DEV_EVENT_DRIVERLOOKAROUND_INFO)
EVENT_IVS_TRAFFIC_DRIVERLEAVEPOST	0x0000020C	开车离岗报警事件(对应 DEV_EVENT_DRIVERLEAVEPOST_INFO)

智能事件名称	序号	含义
EVENT_IVS_MAN_STAND_DETECTION	0x0000020D	立体视觉站立事件(对应 DEV_EVENT_MANSTAND_DETECTION_INFO)
EVENT_IVS_MAN_NUM_DETECTION	0x0000020E	立体视觉区域内人数统计事件(对应 DEV_EVENT_MANNUM_DETECTION_INFO)
EVENT_IVS_STEREO_NUMBERSTAT	0x0000020F	客流量统计事件(暂未有具体事件)
EVENT_IVS_TRAFFIC_DRIVERYAWN	0x00000210	开车打哈欠事件(对应 DEV_EVENT_DRIVERYAWN_INFO)
EVENT_IVS_NUMBERSTAT_PLAN	0x00000211	客流量统计计划(暂未有具体事件,球机使用,对应规则配置结构体 CFG_NUMBERSTAT_INFO)
EVENT_IVS_HEATMAP_PLAN	0x00000212	热度图计划(暂未有具体事件,球机使用,对应规则配置结构体 CFG_IVS_HEATMAP_INFO)
EVENT_IVS_CALLNOANSWERED	0x00000213	呼叫无答应事件
EVENT_IVS_IGNOREINVITE	0x00000214	无视邀请事件
EVENT_IVS_HUMANTRAIT	0x00000215	人体特征事件(对应 DEV_EVENT_HUMANTRAIT_INFO)
EVENT_ALARM_LE_HEADDETECTION	0x00000216	乐橙人头检测事件(对应 DEV_EVENT_LE_HEADDETECTION_INFO)
EVENT_IVS_FACEANALYSIS	0x00000217	目标分析事件(暂未有具体事件)
EVENT_IVS_TRAFFIC_TURNLEFTAFTERSTRAIGHT	0x00000218	左转不礼让直行事件(对应 DEV_EVENT_TURNLEFTAFTERSTRAIGHT_INFO)
EVENT_IVS_TRAFFIC_BIGBENDSMALLTURN	0x00000219	大弯小转事件(对应 DEV_EVENT_BIGBENDSMALLTURN_INFO)
EVENT_IVS_ROAD_CONSTRUCTION	0x0000021A	道路施工监测事件(对应 DEV_EVENT_ROAD_CONSTRUCTION_INFO)
EVENT_IVS_ROAD_BLOCK	0x0000021B	路障检测事件(对应 DEV_EVENT_ROAD_BLOCK_INFO)
EVENT_IVS_TRAFFIC_QUEUEJUMP	0x0000021C	车辆加塞事件(对应 DEV_EVENT_TRAFFIC_QUEUEJUMP_INFO)
EVENT_IVS_VEHICLE_SUSPICIOUSCAR	0x0000021D	嫌疑车辆事件(对应 DEV_EVENT_VEHICLE_SUSPICIOUSCAR_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_TURNRIGHTAFTERSTRAIGHT	0x0000021E	右转不礼让直行事件(对应 DEV_EVENT_TURNRIGHTAFTERSTRAIGHT_INFO)
EVENT_IVS_TRAFFIC_TURNRIGHTAFTERPEOPLE	0x0000021F	右转不礼让直行行人(对应 DEV_EVENT_TURNRIGHTAFTERPEOPLE_INFO)
EVENT_IVS_INSTALL_CARDREADER	0x00000220	安装读卡器事件(对应 DEV_EVENT_INSTALL_CARDREADER_INFO)
EVENT_ALARM_YALE_DROPBOX_BADTOKEN	0x00000221	Yaletoken 失效事件，只用于订阅手机推送
EVENT_IVS_ACC_OFF_SNAP	0x00000222	车载设备断电前抓拍上传事件(对应 DEV_EVENT_ACC_OFF_SNAP_INFO)
EVENI_IVS_XRAY_DETECTION	0x00000223	X 光检测事件(对应 DEV_EVENT_XRAY_DETECTION_INFO)
EVENT_IVS_NOTCLEARCAR	0x00000224	未清车告警(对应 DEV_EVENT_NOTCLEARCAR_INFO)
EVENT_IVS_SOSALEART	0x00000225	sos 求救报警(对应 DEV_EVENT_SOSALEART_INFO)
EVENT_IVS_OVERLOAD	0x00000226	超载抓图(对应 DEV_EVENT_OVERLOAD_INFO)
EVENT_IVS_NONWORKINGTIME	0x00000227	非工作时间告警(对应 DEV_EVENT_NONWORKINGTIME_INFO)
EVENT_IVS_TRAFFIC_HIGH_BEAM	0x00000228	远光灯违章事件(对应 DEV_EVENT_TRAFFIC_HIGH_BEAM_INFO)
EVENT_IVS_TRAFFIC_TRUCKFORBID	0x00000229	禁止货车事件(对应 DEV_EVENT_TRAFFICTRUCKFORBID_INFO)
EVENT_IVS_DRIVINGWITHOUTCARD	0x0000022A	无卡驾驶报警事件(对应 DEV_EVENT_DRIVINGWITHOUTCARD_INFO)
EVENT_IVS_HIGHSPEED	0x0000022B	车辆超速报警事件(对应 DEV_EVENT_HIGHSPEED_INFO)
EVENT_IVS_CROWDDETECTION	0x0000022C	人群密度检测事件(对应结构体 DEV_EVENT_CROWD_DETECTION_INFO)
EVENT_IVS_TRAFFIC_CARDISTANCESHORT	0x0000022D	车间距过小报警事件(对应 DEV_EVENT_TRAFFIC_CARDISTANCESHORT_INFO)

智能事件名称	序号	含义
EVENT_IVS_PEDESTRIAN_JUNCTION	0x00000230	行人卡口事件(对应 DEV_EVENT_PEDESTRIAN_JUNCTION_INFO)
EVENT_IVS_VEHICLE_RECOGNITION	0x00000231	车牌对比事件(对应 DEV_EVENT_VEHICLE_RECOGNITION_INFO)
EVENT_IVS_PASS_CHANGE	0x00000232	预置点图片变化事件(对应 DEV_EVENT_PASS_CHANGE_INFO)
EVENT_IVS_TRAFFIC_PARKING_SPACEDETECTION	0x00000233	违停相机单球车位检测规则事件
EVENT_IVS_TRAFFIC_WAITINGAREA	0x00000234	违章进入待行区事件(对应 DEV_EVENT_TRAFFIC_WAITINGAREA_INFO)
EVENT_IVS_TRAFFIC_BAN	0x00000235	机动车违法禁令标识事件(对应 DEV_EVENT_TRAFFIC_BAN_INFO)
EVENT_IVS_POS_EXCHANGE	0x00000236	POS 机交易事件(对应 DEV_EVENT_POS_EXCHANGE_INFO)
EVENT_IVS_STEREO_FIGHTDETECTION	0x00000237	立体行为分析打架/剧烈运动检测规则(仅用于规则配置，对应事件 EVENT_IVS_FIGHTDETECTION)
EVENT_IVS_STEREO_DISTANCE_DETECTION	0x00000238	立体行为分析间距异常/人员靠近检测(仅用于规则配置，对应事件)
EVENT_IVS_STEREO_STEREOFALLDETECTION	0x00000239	立体行为分析跌倒检测规则(仅用于规则配置，对应事件 EVENT_IVS_TUMBLE_DETECTION)
EVENT_IVS_STEREO_STAYDETECTION	0x0000023A	立体行为分析人员滞留检测规则(仅用于规则配置，对应事件 EVENT_IVS_STAYDETECTION)
EVENT_IVS_BANNER_DETECTION	0x0000023B	拉横幅事件(对应 DEV_EVENT_BANNER_DETECTION_INFO)
EVENT_IVS_NORMAL_FIGHTDETECTION	0x0000023C	普通打架事件(只用于普通打架规则,事件采用 EVENT_IVS_FIGHTDETECTION)
EVENT_IVS_ELEVATOR_ABNORMAL	0x0000023D	电动扶梯运行异常事件(对应 DEV_EVENT_ELEVATOR_ABNORMAL_INFO)
EVENT_IVS_NONMOTORDETECT	0x0000023E	非机动车检测(对应结构体 DEV_EVENT_NONMOTORDETECT_INFO)

智能事件名称	序号	含义
EVENT_IVS_VEHICLEDETECT	0x0000023F	机动车检测(只用于规则配置，事件采用 EVENT_IVS_TRAFFICJUNCTION)
EVENT_IVS_TRAFFIC_PARKING_B	0x00000240	交通违章-B 类违章停车(对应 DEV_EVENT_TRAFFIC_PARKING_B_INFO)
EVENT_IVS_TRAFFIC_PARKING_C	0x00000241	交通违章-C 类违章停车(对应 DEV_EVENT_TRAFFIC_PARKING_C_INFO)
EVENT_IVS_TRAFFIC_PARKING_D	0x00000242	交通违章-D 类违章停车(对应 DEV_EVENT_TRAFFIC_PARKING_D_INFO)
EVENT_IVSS_FACEATTRIBUTE	0x00000243	IVSS 目标检测事件(暂未有具体事件)
EVENT_IVSS_FACECOMPARE	0x00000244	IVSS 目标识别事件(暂未有具体事件)
EVENT_IVS_FIREWARNING	0x00000245	火警事件(对应 DEV_EVENT_FIREWARNING_INFO)
EVENT_IVS_SHOPPRESENCE	0x00000246	商铺占道经营事件(对应 DEV_EVENT_SHOPPRESENCE_INFO)
EVENT_IVS_WASTEDUMPED	0x00000247	垃圾违章倾倒事件(对应 DEV_EVENT_WASTEDUMPED_INFO)
EVENT_IVS_SPILLEDMATERIAL_DETECTION	0x00000248	抛洒物检测事件(对应 DEV_EVENT_SPILLEDMATERIAL_DETECTION_INFO)
EVENT_IVS_STEREO_MANNUM_DETECTION	0x00000249	立体行为分析人数异常检测(仅用于规则配置，对应事件 EVENT_IVS_MAN_NUM_DETECTION)
EVENT_IVS_DISTANCE_DETECTION	0x0000024A	异常间距事件(对应 DEV_EVENT_DISTANCE_DETECTION_INFO)
EVENT_IVS_TRAFFIC_NONMOTOR_OVERLOAD	0x0000024B	非机动车超载事件(对应 DEV_EVENT_TRAFFIC_NONMOTOR_OVERLOAD_INFO)
EVENT_IVS_TRAFFIC_NONMOTOR_WITHOUTSAFEHAT	0x0000024C	非机动车未戴安全帽事件(对应 DEV_EVENT_TRAFFIC_NONMOTOR_WITHOUTSAFEHAT_INFO)
EVENT_IVS_TRAFFIC_JAM_STOP_ON_ZEBRACROSSING	0x0000024D	拥堵滞留斑马线事件(对应 DEV_EVENT_TRAFFIC_JAM_STOP_ON_ZEBRACROSSING_INFO)

智能事件名称	序号	含义
EVENT_IVS_FLOWBUSINESS	0x0000024E	流动摊贩事件(对应 DEV_EVENT_FLOWBUSINESS_INFO)
EVENT_IVS_CITY_MOTORPARKING	0x0000024F	城市机动车违停事件(对应 DEV_EVENT_CITY_MOTORPARKING_INFO)
EVENT_IVS_CITY_NONMOTORPARKING	0x00000250	城市机非动车违停事件(对应 DEV_EVENT_CITY_NONMOTORPARKING_INFO)
EVENT_IVS_LANEDEPARTURE_WARNNING	0x00000251	车道偏移预警(对应 DEV_EVENT_LANEDEPARTURE_WARNNING_INFO)
EVENT_IVS_FORWARDCOLLISION_WARNNING	0x00000252	前向碰撞预警(对应 DEV_EVENT_FORWARDCOLLISION_WARNNING_INFO)
EVENT_IVS_MATERIALSSTAY	0x00000253	物料堆放事件(对应 DEV_EVENT_MATERIALSSTAY_INFO)
EVENT_IVS_TRAFFIC_NONMOTOR_HOLDUMBRELLA	0x00000254	非机动车装载伞具(对应 DEV_EVENT_TRAFFIC_NONMOTOR_HOLDUMBRELLA_INFO)
EVENT_IVS_JABLOTRON_ALARM	0x00000255	客户报警产品
EVENT_IVS_VIDEOUNFOCUS_ALARM	0x00000256	视频虚焦事件(对应 DEV_EVENT_VIDEOUNFOCUS_INFO)
EVENT_IVS_FLOATINGOBJECT_DETECTION	0x00000257	漂浮物检测事件(对应 DEV_EVENT_FLOATINGOBJECT_DETECTION_INFO)
EVENT_IVS_SHIP_DETECTION	0x00000258	船舶检测事件(对应 DEV_EVENT_SHIP_DETECTION_INFO)
EVENT_IVS_AIRPLANE_DETECTION	0x00000259	飞机行为检测事件(对应 DEV_EVENT_AIRPLANE_DETECTION_INFO)
EVENT_IVS_PHONECALL_DETECT	0x0000025A	打电话检测事件(对应 DEV_EVENT_PHONECALL_DETECT_INFO)
EVENT_IVS_SMOKING_DETECT	0x0000025B	吸烟检测事件(对应 DEV_EVENT_SMOKING_DETECT_INFO)
EVENT_IVS_RADAR_SPEED_LIMIT_ALARM	0x0000025C	雷达限速报警事件(对应 DEV_EVENT_RADAR_SPEED_LIMIT_ALARM_INFO)

智能事件名称	序号	含义
EVENT_IVS_WATER_LEVEL_DETECTION	0x0000025D	水位检测事件(对应 DEV_EVENT_WATER_LEVEL_DETECTION_INFO)
EVENT_IVS_HOLD_UMBRELLA	0x0000025E	违规撑伞检测事件(对应 DEV_EVENT_HOLD_UMBRELLA_INFO)
EVENT_IVS_GARBAGE_EXPOSURE	0x0000025F	垃圾暴露检测事件(对应 DEV_EVENT_GARBAGE_EXPOSURE_INFO)
EVENT_IVS_DUSTBIN_OVER_FLOW	0x00000260	垃圾桶满溢检测事件(对应 DEV_EVENT_DUSTBIN_OVER_FLOW_INFO)
EVENT_IVS_DOOR_FRONT_DIRTY	0x00000261	门前脏乱检测事件(对应 DEV_EVENT_DOOR_FRONT_DIRTY_INFO)
EVENT_IVS_QUEUESTAY_DETECTION	0x00000262	排队滞留时间报警事件(对应 DEV_EVENT_QUEUESTAY_DETECTION_INFO)
EVENT_IVS_QUEUENUM_DETECTION	0x00000263	排队人数异常报警事件（对应 DEV_EVENT_QUEUENUM_DETECTION_INFO）
EVENT_IVS_GENERATEGRAPH_DETECTION	0x00000264	生成图规则事件（对应 DEV_EVENT_GENERATEGRAPH_DETECTION_INFO）
EVENT_IVS_TRAFFIC_PARKING_MANUAL	0x00000265	交通违章-手动取证(对应 DEV_EVENT_TRAFFIC_PARKING_MANUAL_INFO)
EVENT_IVS_HELMET_DETECTION	0x00000266	安全帽检测事件(对应 DEV_EVENT_HELMET_DETECTION_INFO)
EVENT_IVS_DEPOSIT_DETECTION	0x00000267	包裹堆积程度检测事件(对应 DEV_EVENT_DEPOSIT_DETECTION_INFO)
EVENT_IVS_HOTSPOT_WARNING	0x00000268	热点异常报警事件(对应 DEV_EVENT_HOTSPOT_WARNING_INFO)
EVENT_IVS_WEIGHING_PLATFORM_DETECTION	0x00000269	称重平台检测事件(对应 DEV_EVENT_WEIGHING_PLATFORM_DETECTION_INFO)
EVENT_IVS_CLASSROOM_BEHAVIOR	0x0000026A	课堂行为分析事件(对应 DEV_EVENT_CLASSROOM_BEHAVIOR_INFO)
EVENT_IVS_VEHICLE_DISTANCE_NEAR	0x0000026B	安全驾驶车距过近报警事件(对应 DEV_EVENT_VEHICLE_DISTANCE_NEAR_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_DRIVER_ABNORMAL	0x0000026C	驾驶员异常报警事件(对应 DEV_EVENT_TRAFFIC_DRIVER_ABNORMAL_INFO)
EVENT_IVS_TRAFFIC_DRIVER_CHANGE	0x0000026D	驾驶员变更报警事件(对应 DEV_EVENT_TRAFFIC_DRIVER_CHANGE_INFO)
EVENT_IVS_WORKCLOTHES_DETECT	0x0000026E	工装(安全帽/工作服等)检测事件(对应 DEV_EVENT_WORKCLOTHES_DETECT_INFO)
EVENT_IVS_SECURITYGATE_PERSONALARM	0x0000026F	安检门人员报警事件(对应 DEV_EVENT_SECURITYGATE_PERSONALARM_INFO)
EVENT_IVS_STAY_ALONE_DETECTION	0x00000270	单人独处事件(对应 DEV_EVENT_STAY_ALONE_DETECTION_INFO)
EVENT_IVS_TRAFFIC_ROAD_BLOCK	0x00000271	交通路障检测事件(对应 DEV_EVENT_TRAFFIC_ROAD_BLOCK_INFO)
EVENT_IVS_TRAFFIC_ROAD_CONSTRUCTION	0x00000272	交通道路施工检测事件(对应 DEV_EVENT_TRAFFIC_ROAD_CONSTRUCTION_INFO)
EVENT_IVS_XRAY_DETECT_BYOBJECT	0x00000273	X 光按物体检测规则配置,对应事件 EVENI_IVS_XRAY_DETECTION
EVENT_IVS_WORKSTATDETECTION	0x00000274	作业统计事件(对应 DEV_EVENT_WORKSTATDETECTION_INFO)
EVENT_IVS_INFRAREDBLOCK	0x00000275	红外阻断事件(对应 DEV_EVENT_INFRAREDBLOCK_INFO)
EVENT_IVS_FEATURE_ABSTRACT	0x00000276	特征提取事件(对应 DEV_EVENT_FEATURE_ABSTRACT_INFO)
EVENT_IVS_INTELLI_SHELF	0x00000277	智能补货事件(对应 DEV_EVENT_INTELLI_SHELF_INFO)
EVENT_IVS_PANORAMA_SHOT	0x00000278	全景抓拍事件(对应 DEV_EVENT_PANORAMA_SHOT_INFO)
EVENT_ALARM_SMARTMOTION_HUMAN	0x00000279	智能视频移动侦测事件(人),(对应 DEV_EVENT_SMARTMOTION_HUMAN_INFO)
EVENT_ALARM_SMARTMOTION_VEHICLE	0x0000027A	智能视频移动侦测事件(车),(对应 DEV_EVENT_SMARTMOTION_VEHICLE_INFO)
EVENT_IVS_CAR_DRIVING_IN_OUT	0x0000027B	车辆驶入驶出状态事件(对应 DEV_EVENT_CAR_DRIVING_IN_OUT_INFO)

智能事件名称	序号	含义
EVENT_IVS_PARKINGSPACE_STATUS	0x0000027C	停车位状态事件(对应 DEV_EVENT_PARKINGSPACE_STATUS_INFO)
EVENT_IVS_VIOLENT_THROW_DETECTION	0x0000027D	暴力抛物检测(对应 DEV_EVENT_VIOLENT_THROW_DETECTION_INFO)
EVENT_IVS_TRAMCARSECTIONS_DETECTION	0x0000027E	矿车超挂报警事件(对应 DEV_EVENT_TRAMCARSECTIONS_DETECTION_INFO)
EVENT_IVS_ALARM_BOX_ALARM	0x0000027F	报警盒通道的触发报警事件,目前只用于订阅手机推送
EVENT_IVS_FACE_COMPARISION	0x00000280	目标比对事件,专用于手机推送的目标识别事件,目前只用于订阅手机推送
EVENT_IVS_FACEBODY_DETECT	0x00000281	人像检测事件（对应 DEV_EVENT_FACEBODY_DETECT_INFO）
EVENT_IVS_FACEBODY_ANALYSE	0x00000282	人像识别事件（对应 DEV_EVENT_FACEBODY_ANALYSE_INFO）
EVENT_IVS_GASSTATION_VEHICLE_DETECT	0x00000283	加油站车辆检测事件(对应 DEV_EVENT_GASSTATION_VEHICLE_DETECT_INFO)
EVENT_IVS_CONGESTION_DETECTION	0x00000284	道路场景车辆拥堵报警事件（对应 DEV_EVENT_CONGESTION_DETECTION_INFO ）
EVENT_IVS_VEHICLELIMIT_DETECTION	0x00000285	停车场场景下停车车辆上限报警（对应 DEV_EVENT_VEHICLELIMIT_DETECTION_INFO ）
EVENT_IVS_ANIMAL_DETECTION	0x00000286	动物检测事件,(对应 DEV_EVENT_ANIMAL_DETECTION_INFO)
EVENT_IVS_SHOP_WINDOW_POST	0x00000287	橱窗张贴事件(对应 DEV_EVENT_SHOP_WINDOW_POST_INFO)
EVENT_IVS_SHOP_SIGN_ABNORMAL	0x00000288	店招异常事件(对应 DEV_EVENT_SHOP_SIGN_ABNORMAL_INFO)
EVENT_IVS_BREED_DETECTION	0x00000289	智慧养殖检测事件(对应 DEV_EVENT_BREED_DETECTION_INFO)
EVENT_IVS_AIRPORT_VEHICLE_DETECT	0x0000028A	机场智能保障车辆检测事件(对应 DEV_EVENT_AIRPORT_VEHICLE_DETECT_INFO)

智能事件名称	序号	含义
EVENT_IVS_PIG_TEMPERATURE_DETECT	0x0000028B	智慧养殖猪体温检测(只用于规则配置)
EVENT_IVS_MAN_CAR_COEXISTANCE	0x0000028C	人车共存事件(对应 DEV_EVENT_MAN_CAR_COEXISTANCE_INFO)
EVENT_IVS_HIGH_TOSS_DETECT	0x0000028D	高空抛物检测(对应 DEV_EVENT_HIGH_TOSS_DETECT_INFO)
EVENT_IVS_ELECTRIC_GLOVE_DETECT	0x0000028E	电力检测手套检测事件(对应 DEV_EVENT_ELECTRIC_GLOVE_DETECT_INFO)
EVENT_IVS_ELECTRIC_LADDER_DETECT	0x0000028F	电力检测梯子检测事件(对应 DEV_EVENT_ELECTRIC_LADDER_DETECT_INFO)
EVENT_IVS_ELECTRIC_CURTAIN_DETECT	0x00000290	电力检测布幔检测事件(对应 DEV_EVENT_ELECTRIC_CURTAIN_DETECT_INFO)
EVENT_IVS_ELECTRIC_FENCE_DETECT	0x00000291	电力检测围栏检测事件(对应 DEV_EVENT_ELECTRIC_FENCE_DETECT_INFO)
EVENT_IVS_ELECTRIC_SIGNBOARD_DETECT	0x00000292	电力检测标识牌检测事件(对应 DEV_EVENT_ELECTRIC_SIGNBOARD_DETECT_INFO)
EVENT_IVS_ELECTRIC_BELT_DETECT	0x00000293	电力检测安全带检测事件(对应 DEV_EVENT_ELECTRIC_BELT_DETECT_INFO)
EVENT_IVS_RADAR_LINE_DETECTION	0x00000294	雷达警戒线/绊线检测(对应 DEV_EVENT_RADAR_LINE_DETECTION_INFO)
EVENT_IVS_RADAR_REGION_DETECTION	0x00000295	雷达警戒区检测事件(对应 DEV_EVENT_RADAR_REGION_DETECTION_INFO)
EVENT_IVS_AUDIO_INTENSITY	0x00000296	异常音事件(对应 DEV_EVENT_AUDIO_INTENSITY_INFO)
EVENT_IVS_PARKING_LOT_STATUS_DETECTION	0x00000297	室外停车位状态检测(对应 DEV_EVENT_PARKING_LOT_STATUS_DETECTION_INFO)
EVENT_IVS_VEHICLE_COMPARE	0x00000298	(只用于规则配置)
EVENT_IVS_DREGS_UNCOVERED	0x00000299	渣土车未遮盖载货检测事件(对应 DEV_EVENT_DREGS_UNCOVERED_INFO)

智能事件名称	序号	含义
EVENT_IVS_WALK_DETECTION	0x0000029A	走动检测事件(对应 DEV_EVENT_WALK_DETECTION_INFO)
EVENT_IVS_BACK_TO_DETECTION	0x0000029B	背对检测事件(对应 DEV_EVENT_BACK_TO_DETECTION_INFO)
EVENT_IVS_WRITE_ON_THE_BOARD_DETECTION	0x0000029C	板书检测事件(对应 DEV_EVENT_WRITE_ON_THE_BOARD_DETECTION_INFO)
EVENT_IVS_SMART_KITCHEN_CLOTHES_DETECTION	0x0000029D	智慧厨房穿着检测事件（对不戴口罩、厨师帽以及颜色不符合规定的厨师服进行报警）（对应 DEV_EVENT_SMART_KITCHEN_CLOTHES_DETECTION_INFO）
EVENT_IVS_SLEEP_DETECT	0x0000029E	睡觉检测事件(对应 DEV_EVENT_SLEEP_DETECT_INFO)
EVENT_IVS_WALK_AROUND_DETECT	0x0000029F	随意走动检测事件(对应 DEV_EVENT_WALK_AROUND_DETECT_INFO)
EVENT_IVS_PLAY_MOBILEPHONE	0x00000300	玩手机事件(对应 DEV_EVENT_PLAY_MOBILEPHONE_INFO)
EVENT_IVS_FINANCE_CONTRABAND_DETECT	0x00000301	智慧金融违规物品检测事件(对应 DEV_EVENT_FINANCE_CONTRABAND_DETECT_INFO)
EVENT_IVS_FINANCE_CASH_TRANSACTION	0x00000302	智慧金融现金交易检测事件(对应 DEV_EVENT_FINANCE_CASH_TRANSACTION_INFO)
EVENT_IVS_ANATOMY_TEMP_DETECT	0x00000303	人体温智能检测事件(对应 DEV_EVENT_ANATOMY_TEMP_DETECT_INFO)
EVENT_IVS_ACTIVITY_ANALYSE	0x00000304	活跃度统计规则(只用于规则配置)
EVENT_IVS_DOOR_STATUS	0x00000305	门状态事件(对应 DEV_EVENT_DOOR_STATUS_INFO)
EVENT_IVS_DHOP_CUSTOM	0x00000306	Dhop 自定义事件(start/stop,对应 DEV_EVENT_DHOP_CUSTOM_INFO)
EVENT_IVS_DHOP_CUSTOM_ONCE	0x00000307	Dhop 自定义事件(Pulse,对应 DEV_EVENT_DHOP_CUSTOM_INFO)

智能事件名称	序号	含义
EVENT_IVS_FOG_DETECTION	0x00000308	起雾检测事件(对应 DEV_EVENT_FOG_DETECTION)
EVENT_IVS_TRAFFIC_VEHICLE_BC	0x00000309	飙车事件（对应 DEV_EVENT_TRAFFIC_VEHICLE_BC）
EVENT_IVS_TRAFFIC_MOTOR_OVERLOAD	0x0000030A	机动车超载(对应 DEV_EVENT_TRAFFIC_MOTOR_OVERLOAD_INFO)
EVENT_IVS_TRAFFIC_PLATE_OCCLUSION	0x0000030B	车牌污损(对应 DEV_EVENT_TRAFFIC_PLATE_OCCLUSION_INFO)
EVENT_IVS_NONMOTOR_ENTRYING	0x0000030C	非机动车进入电梯(对应 DEV_EVENT_NONMOTOR_ENTRYING_INFO)
EVENT_IVS_WATER_STAGE_MONITOR	0x0000030D	水位监测事件,目前仅用于任务型智能分析(对应 DEV_EVENT_WATER_STAGE_MONITOR_INFO)
EVENT_IVS_TRAFFIC_ROAD_ALERT	0x0000030E	道路安全预警(对应 DEV_EVENT_TRAFFIC_ROAD_ALERT_INFO)
EVENT_IVS_BREAK_RULE_BUILDING_DETECTION	0x0000030F	违章建筑检测事件(对应 DEV_EVENT_BREAK_RULE_BUILDIING_DETECTION_INFO)
EVENT_IVS_TRAFFIC_NONMOTOR_RUN_REDLIGHT	0x00000310	非机动车闯红灯(对应 DEV_EVENT_TRAFFIC_NONMOTOR_RUN_REDLIGHT_INFO)
EVENT_IVS_TRAFFIC_VEHICLE_IN_EMERGENCY_LANE	0x00000311	占用应急车道事件(对应 DEV_EVENT_TRAFFIC_VEHICLE_IN_EMERGENCY_LANE_INFO)
EVENT_IVS_PRAM_DETECTION	0x00000312	婴儿车检测事件(对应 DEV_EVENT_PRAM_DETECTION_INFO)
EVENT_IVS_STEREO_PRAM_DETECTION	0x00000313	立体行为婴儿车检测事件(只用于规则配置)
EVENT_IVS_BIG_BAGGAGE_DETECTION	0x00000314	大件行李箱检测事件(对应 DEV_EVENT_BIG_BAGGAGE_DETECTION_INFO)
EVENT_IVS_STEREO_BIG_BAGGAGE_DETECTION	0x00000315	立体行为大件行李箱检测事件(只用于规则配置)

智能事件名称	序号	含义
EVENT_IVS_TICKET_EVADE_DETECTION	0x00000316	逃票检测事件(对应 DEV_EVENT_TICKET_EVADE_DETECTION_INFO)
EVENT_IVS_STEREO_TICKET_EVADE_DETECTION	0x00000317	立体行为逃票检测事件(只用于规则配置)
EVENT_IVS_POWERLINE_FOREIGN_DETECITON	0x00000318	输电线异物检测(对应 DEV_EVENT_POWERLINE_FOREIGN_DETEC_INFO)
EVENT_IVS_TRAFFIC_OVER_GUIDE_LINE	0x00000319	压导流线(对应 DEV_EVENT_TRAFFIC_OVER_GUIDE_LINE_INFO)
EVENT_IVS_TRAFFIC_CAR_MEASUREMENT	0x00000320	交通卡口测量(车辆长、宽、高度、重量等)事件(对应 DEV_EVENT_TRAFFIC_CAR_MEASUREMENT_INFO)
EVENT_IVS_TRAFFIC_WRONG_TURN_LIGHT	0x00000321	不安规定使用转向灯事件（对应 DEV_EVENT_TRAFFIC_WRONG_TURN_LIGHT_INFO）
EVENT_IVS_TRAFFIC_REAREND_ACCIDENT	0x00000322	交通事故事件（对应 DEV_EVENT_TRAFFIC_REAREND_ACCIDENT_INFO）
EVENT_IVS_DO_TALK_ACTION	0x00000323	对讲动作事件（对应 DEV_EVENT_DO_TALK_ACTION_INFO）
EVENT_IVS_FIRE_LANE_DETECTION	0x00000324	消防占道检测事件（对应 DEV_EVENT_FIRE_LANE_DETECTION_INFO）
EVENT_IVS_PARKING_DETECTION_FOR_PRMA	0x00000325	全景异常停车事件(只用于规则配置)
EVENT_IVS_TRAFFIC_JAM_FOR_PRMA	0x00000326	全景交通拥堵事件(只用于规则配置)
EVENT_IVS_TRAFFIC_ACCIDENT_FOR_PRMA	0x00000327	全景交通事故事件(只用于规则配置)
EVENT_IVS_TRAFFIC_NON_MOTOR_RETROGRADE	0x00000328	非机动车逆行事件(对应 DEV_EVENT_TRAFFIC_NON_MOTOR_RETROGRADE_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_NON_MOTOR_OVER_STOP_LINE	0x00000329	非机动车越线停车事件(对应 DEV_EVENT_TRAFFIC_NON_MOTOR_OVER_STOP_LINE_INFO)
EVENT_IVS_CAR_DRIVING_IN	0x00000330	车辆驶入事件(对应 DEV_EVENT_CAR_DRIVING_IN_INFO)
EVENT_IVS_CAR_DRIVING_OUT	0x00000331	车辆驶出事件(对应 DEV_EVENT_CAR_DRIVING_OUT_INFO)
EVENT_IVS_PORTRAIT_DETECTION	0x00000332	人像检测(只用于规则配置)
EVENT_IVS_TRAFFIC_SPECIAL_VEHICLE_DETECT	0x00000333	特殊车辆检测(对应 DEV_EVENT_TRAFFIC_SPECIAL_VEHICLE_INFO)
EVENT_IVS_TRAFFIC_HEAD_LAMP_OFF	0x00000334	未开照明灯事件(对应 DEV_EVENT_TRAFFIC_HEAD_LAMP_OFF_INFO)
EVENT_IVS_TRAFFIC_NONMOTOR	0x00000335	交通非机动车事件检测,智能服务器使用 (DEV_EVENT_TRAFFIC_NONMOTOR_INFO)
EVENT_IVS_TRAFFIC_BOARD	0x00000336	交通违章上下客事件检测(对应 DEV_EVENT_TRAFFIC_BOARD_INFO)
EVENT_IVS_TRAFFIC_VISIBILITY	0x00000337	交通能见度事件检测(对应 DEV_EVENT_TRAFFIC_VISIBILITY_INFO)
EVENT_IVS_TRAFFIC_VEHICLE_CLEANLINESS	0x00000338	交通车辆清洁度检测事件检测(对应 DEV_EVENT_TRAFFIC_VEHICLE_CLEANLINESS_INFO)
EVENT_IVS_TRAFFICFLOW_FOR_PRMA	0x00000339	全景交通车流(只用于规则配置)
EVENT_IVS_TRUCKNOTCLEAN_FOR_PRMA	0x0000033A	工程车未清洗对应 DEV_EVENT_TRUCKNOTCLEAN_FOR_PRMA_INFO
EVENT_IVS_ROADOCCUPATION_BY_FOREIGNOBJECT	0x0000033B	异物占道事件（对应 DEV_EVENT_ROADOCCUPATION_BY_FOREIGNOBJECT_INFO）
EVENT_IVS_TRAFFICFLOW_OVER	0x0000033C	车流量超过上限(对应 DEV_EVENT_TRAFFICFLOW_OVER_INFO)
EVENT_IVS_GOODS_DETECTION	0x0000033D	违规物品检测事件(对应 DEV_EVENT_GOODS_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_CONVEYORBLOCK_DETECTION	0x0000033E	传送带阻塞报警事件（对应 DEV_EVENT_CONVEYORBLOCK_DETECTION_INFO）
EVENT_IVS_ANYTHING_DETECT	0x0000033F	全物体类型检测事件(对应 DEV_EVENT_ANYTHING_DETECT_INFO)
EVENT_IVS_OBJECT_ABNORMAL	0x00000340	目标异常事件(对应 DEV_EVENT_OBJECT_ABNORMAL_INFO)
EVENT_IVS_DRIVE_ASSISTANT	0x00000341	辅助驾驶(只用于规则配置)
EVENT_IVS_DRIVE_ACTION_ANAYLSE	0x00000342	驾驶行为分析(只用于规则配置)
EVENT_IVS_DRIVE_HANDSOFF_STEERING_WHEEL	0x00000343	驾驶行为分析手离开方向盘检测(对应 DEV_EVENT_DRIVE_HANDSOFF_STEERING_WHEEL_INFO)
EVENT_IVS_DRIVE_BLIND_SPOT	0x00000344	驾驶行为分析盲点检测(对应 DEV_EVENT_BLIND_SPOT_DETECTION_INFO)
EVENT_IVS_ARTICLE_DETECTION	0x00000345	物品检测(只用于规则配置),对应事件 EVENT_IVS_LEFTDETECTION 或者 EVENT_IVS_TAKENAWAYDETECTION
EVENT_IVS_TRAFFIC_PARKINGSPACE_MANUALSNAP	0x00000346	路侧停车位手动抓图(对应 DEV_EVENT_PARKINGSPACE_MANUALSNAP_INFO)
EVENT_IVS_STREET_SUNCURE	0x00000347	沿街晾晒事件（对应 DEV_EVENT_STREET_SUNCURE_INFO）
EVENT_IVS_OUTDOOR_ADVERTISEMENT	0x00000348	户外广告事件（对应 DEV_EVENT_OUTDOOR_ADVERTISEMENT_INFO）
EVENT_IVS_HUDDLE_MATERIAL	0x00000349	乱堆物料检测事件（对应 DEV_EVENT_HUDDLE_MATERIAL_INFO）
EVENT_IVS_FIRE_LINE_DETECTION	0x0000034A	进入消防通道检测(对应 DEV_EVENT_FIRE_LINE_DETECTION_INFO)
EVENT_IVS_OCCUPY_BUS_LANE	0x0000034B	违法占用公交车道（对应 DEV_EVENT_OCCUPY_BUS_LANE_INFO）
EVENT_IVS_DISTRESS_DETECTION	0x0000034C	求救检测事件(对应 DEV_EVENT_DISTRESS_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_ASSISTANT_WITHOUT_SAFEBELT	0x0000034D	交通副驾驶未系安全带事件(对应 DEV_EVENT_TRAFFIC_ASSISTANT_WITHOUT_SAFEBELT)
EVENT_IVS_TRAFFIC_SPEED_CHANGE_DETECTION	0x0000034E	交通变速检测事件(对应 DEV_EVENT_TRAFFIC_SPEED_CHANGE_DETECTION_INFO)
EVENT_IVS_FOLLOW_CAR_ALARM	0x0000034F	跟车报警(对应 DEV_EVENT_FOLLOW_CAR_ALARM_INFO)
EVENT_IVS_CONVEYER_BELT_DETECT	0x00000350	传送带检测,仅用于规则配置,对应规则结构体 CFG_CONVEYER_BELT_DETECT_INFO,对应事件 EVENT_IVS_CONVEYER_BELT_BULK/EVENT_IVS_CONVEYER_BELT_NONLOAD/EVENT_IVS_CONVEYER_BELT_RUNOFF/EVENT_IVS_CONVEYORBLOCK_DETECTION/EVENT_IVS_CONVEYER_BELT_COAL_RATIO
EVENT_IVS_CONVEYER_BELT_BULK	0x00000351	传送带大块异物检测事件(对应 DEV_EVENT_CONVEYER_BELT_BULK_INFO)
EVENT_IVS_CONVEYER_BELT_NONLOAD	0x00000352	传送带空载检测事件(对应 DEV_EVENT_CONVEYER_BELT_NONLOAD_INFO)
EVENT_IVS_CONVEYER_BELT_RUNOFF	0x00000353	传送带跑偏检测事件(对应 DEV_EVENT_CONVEYER_BELT_RUNOFF_INFO)
EVENT_IVS_CONVEYER_BELT_COAL_RATIO	0x00000354	传送带煤量检测(对应 DEV_EVENT_CONVEYER_BELT_COAL_RATIO_INFO)
EVENT_IVS_PACKBROKEN_DETECTION	0x00000355	包裹破损检测事件(对应 DEV_EVENT_PACKBROKEN_DETECTION_INFO)
EVENT_IVS_PACKLAND_DETECTION	0x00000356	包裹着地检测事件(对应 DEV_EVENT_PACKLAND_DETECTION_INFO)
EVENT_IVS_PACKOPEN_DETECTION	0x00000357	开箱验示检测事件(对应 DEV_EVENT_PACKOPEN_DETECTION_INFO)
EVENT_IVS_TRAFFIC_TURN_RIGHT_NO_STOP	0x00000358	右转不停再起步事件(对应 DEV_EVENT_TRAFFIC_TURN_RIGHT_NO_STOP_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_CROSSING_GUARDRAIL	0x00000359	跨越护栏事件(对应 DEV_EVENT_TRAFFIC_CROSSING_GUARDRAIL_INFO)
EVENT_IVS_EMERGENCY	0x0000035A	人为触发的紧急事件，一般处理是联动外部通讯功能请求帮助(对应 DEV_EVENT_EMERGENCY_INFO)
EVENT_IVS_TRAFFIC_PARKING_STATISTICS	0x0000035B	车位统计事件（对应 DEV_EVENT_TRAFFIC_PARKING_STATISTICS_INFO）
EVENT_IVS_HEAT_IMAGING_TEMPER	0x0000035C	热成像测温点温度异常报警事件(对应结构体 DEV_EVENT_HEAT_IMAGING_TEMPER_INFO)
EVENT_IVS_SCALPER_ALARM	0x0000035D	黄牛高频报警事件（对应 DEV_EVENT_SCALPER_ALARM_INFO）
EVENT_IVS_ROAD_OBSTACLE_DETECTION	0x0000035E	道路障碍物检测事件(对应 DEV_EVENT_ROAD_OBSTACLE_DETECTION_INFO)
EVENT_IVS_TRAFFIC_BRIGANDAGE_CAR	0x0000035F	盗抢车辆事件（对应 DEV_EVENT_TRAFFIC_BRIGANDAGE_CAR_INFO）
EVENT_IVS_TRAFFIC_COUNTERFEIT_PLATE_CAR	0x00000360	假牌车辆事件（对应 DEV_EVENT_TRAFFIC_COUNTERFEIT_PLATE_CAR_INFO）
EVENT_IVS_TRAFFIC_FAKE_PLATE_CAR	0x00000361	套牌车辆事件（对应 DEV_EVENT_TRAFFIC_FAKE_PLATE_CAR_INFO ）
EVENT_IVS_SEWAGE_DETECTION	0x00000362	排污检测事件（对应 DEV_EVENT_SEWAGE_DETECTION_INFO)
EVENT_IVS_WATERCOLOR_DETECTION	0x00000363	水体颜色事件（对应 DEV_EVENT_WATERCOLOR_DETECTION_INFO ）
EVENT_IVS_TRAFFIC_MOTORCYCLE_FORBID	0x00000364	禁摩事件(对应 DEV_EVENT_TRAFFIC_MOTORCYCLE_FORBID)

智能事件名称	序号	含义
EVENT_IVS_VIDEO_NORMAL_DETECTION	0x00000365	视频正常事件,在视频诊断检测周期结束时,将未报错的诊断项上报正常事件 DEV_EVENT_VIDEO_NORMAL_DETECTION_INFO
EVENT_IVS_TRAFFIC_TURN_RIGHT_OVER_LINE	0x00000366	右转弯压线事件(对应 DEV_EVENT_TRAFFIC_TURN_RIGHT_OVER_LINE_INFO)
EVENT_IVS_MANUAL_ALARM	0x00000367	人工报警事件(对应 DEV_EVENT_MANUAL_ALARM_INFO)
EVENT_IVS_TRAFFIC_DRIVE_ON_LINE	0x00000368	骑线行驶事件(对应 DEV_EVENT_TRAFFIC_DRIVE_ON_LINE_INFO)
EVENT_IVS_OBJECT_PLACEMENT_DETECTION	0x00000369	物品放置检测事件(对应 DEV_EVENT_OBJECT_PLACEMENT_DETECTION_INFO)
EVENT_IVS_OBJECT_REMOVAL_DETECTION	0x0000036A	物品拿取检测事件(对应 DEV_EVENT_OBJECT_REMOVAL_DETECTION_INFO)
EVENT_IVS_FIRE_DOOR_DETECTION	0x0000036B	防火门检测事件(对应 DEV_EVENT_FIRE_DOOR_DETECTION_INFO)
EVENT_IVS_FIRE_EXTINGUISHER_DETECTION	0x0000036C	灭火器检测事件(对应 DEV_EVENT_FIRE_EXTINGUISHER_DETECTION_INFO)
EVENT_IVS_DOOR_NOT_CLOSE	0x0000036D	门未关事件(对应 DEV_EVENT_DOOR_NOT_CLOSE_INFO)
EVENT_IVS_VEHICLE_PERIPHERAL_ALARM	0x0000036E	车载外设异常报警事件(对应 DEV_EVENT_VEHICLE_PERIPHERAL_ALARM_INFO)
EVENT_IVS_PARKING_LIMIT_DETECTION	0x0000036F	车位阈值变化报警(对应 DEV_EVENT_PARKING_LIMIT_DETECTION_INFO)
EVENT_IVS_PARKING_STATUS_CHANGE_DETECTION	0x00000370	车位状态变化报警(对应 DEV_EVENT_PARKING_STATUS_CHANGE_DETECTION_INFO)
EVENT_IVS_DIALRECOGNITION	0x00000371	仪表检测事件(仅用于规则配置,对应 DEV_EVENT_DIALRECOGNITION_INFO)

智能事件名称	序号	含义
EVENT_IVS_ELECTRICFAULT_DETECT	0x00000372	仪表类缺陷检测事件(对应 DEV_EVENT_ELECTRICFAULTDETECT_INFO)
EVENT_IVS_TRASH_WITHOUT_COVER_DETECTION	0x00000373	垃圾桶未盖盖子检测事件(对应 DEV_EVENT_TRASH_WITHOUT_COVER_DETECTION_INFO)
EVENT_IVS_SECURITY_INSPECTOR_LOOKAROUND	0x00000374	安检员左顾右盼报警事件(对应 DEV_EVENT_SECURITY_INSPECTOR_LOOKAROUND_INFO)
EVENT_IVS_SECURITY_INSPECTOR_LOWERHEAD	0x00000375	安检员低头报警事件(对应 DEV_EVENT_SECURITY_INSPECTOR_LOWERHEAD_INFO)
EVENT_IVS_SECURITY_INSPECTOR_TIRED	0x00000376	安检员疲劳报警事件(对应 DEV_EVENT_SECURITY_INSPECTOR_TIRED_INFO)
EVENT_IVS_SECURITY_INSPECTOR_YAWN	0x00000377	安检员打哈欠报警事件(对应 DEV_EVENT_SECURITY_INSPECTOR_YAWN_INFO)
EVENT_IVS_AIPICK_DETECT	0x00000378	智能拾音器语音识别事件(对应 DEV_EVENT_AIPICK_DETECT_INFO)
EVENT_IVS_TRAFFIC_LIGHT_FAULT	0x00000379	交通灯故障报警事件(对应 DEV_EVENT_TRAFFIC_LIGHT_FAULT_INFO)
EVENT_IVS_TRAFFIC_LANE_INDICATOR_FAULT	0x0000037A	交通变道指示牌故障事件(对应 DEV_EVENT_TRAFFIC_LANE_INDICATOR_FAULT_INFO)
EVENT_IVS_WATER_SPEED_DETECTION	0x0000037B	水流速检测事件(对应 DEV_EVENT_WATER_SPEED_DETECTION_INFO)
EVENT_IVS_TRAFFIC_PARKING_BACKING	0x0000037C	出入口倒车驶离事件(对应 DEV_EVENT_TRAFFIC_PARKING_BACKING_INFO)
EVENT_IVS_VALVE_ABNORMAL	0x0000037D	阀门异常检测事件(对应 DEV_EVENT_VALVE_ABNORMAL_INFO)
EVENT_IVS_DISCHARGE_PORT_ABNORMAL	0x0000037E	出料口异常检测事件(对应 DEV_EVENT_DISCHARGE_PORT_ABNORMAL_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_MOTOR_STRICTLY_PROHIBIT_MANNED	0x0000037F	违法裁人事件(对应 DEV_EVENT_TRAFFIC_MOTOR_STRICTLY_PROHIBIT_MANNED_INFO)
EVENT_IVS_BARELAND_DETECTION	0x00000380	裸土检测事件 (DEV_EVENT_BARELAND_DETECTION_INFO)
EVENT_IVS_CONSUMPTION_EVENT	0x00000381	消费事件(对应 DEV_EVENT_CONSUMPTION_EVENT_INFO)
EVENT_IVS_TOUCH_ELECTROSTATIC_BALL	0x00000382	触摸静电球事件(对应 DEV_EVENT_TOUCH_ELECTROSTATIC_BALL_INFO)
EVENT_IVS_OXYGEN_CYLINDER_DETECTION	0x00000383	氧气瓶检测事件(对应 DEV_EVENT_OXYGEN_CYLINDER_DETECTION_INFO)
EVENT_IVS_XRAY_UNPACKING_CHECK	0x00000384	X 光开包检查事件(对应 DEV_EVENT_XRAY_UNPACKING_CHECK_INFO)
EVENT_IVS_GENERAL_ATTITUDE_DETECTION	0x00000385	姿态检测事件(对应 DEV_EVENT_GENERAL_ATTITUDE_DETECTION_INFO)
EVENT_IVS_SNAP_TASK	0x00000386	非接入式任务抓图事件(对应 DEV_EVENT_SNAP_TASK_INFO)
EVENT_IVS_TRAFFIC_CHANGE_LANE_CONTINUES	0x00000387	机动车连续变道违法事件(对应 DEV_EVENT_TRAFFIC_CHANGE_LANE_CONTINUES_INFO)
EVENT_IVS_SPRAY_DETECT	0x00000388	喷洒异常检测事件(对应 DEV_EVENT_SPRAY_DETECT_INFO)
EVENT_IVS_SILICON_FIRE_DETECTION	0x00000389	刺火检测事件(对应 DEV_EVENT_SILICON_FIRE_DETECTION_INFO)
EVENT_IVS_TRAFFIC_PARKING_ON_RIGHT_TURN_ROUTE	0x0000038A	右转道违停(对应 DEV_EVENT_TRAFFIC_PARKING_ON_RIGHT_TURN_ROUTE_INFO)
EVENT_IVS_PERSONNEL_COEXISTENCE	0x0000038B	人员共存事件(对应 DEV_EVENT_PERSONNEL_COEXISTENCE_INFO)
EVENT_IVS_GENEAL_ATTITUDE	0x0000038C	通用姿态行为事件(对应 DEV_EVENT_GENEAL_ATTITUDE_INFO)

智能事件名称	序号	含义
EVENT_IVS_CASH_BOX_STATE	0x0000038D	款箱状态事件(对应 DEV_EVENT_CASH_BOX_STATE_INFO)
EVENT_IVS_LEAKAGE_DETECTION	0x0000038E	渗漏检测事件(对应 DEV_EVENT_LEAKAGE_DETECTION_INFO)
EVENT_IVS_LADLE_NO_DETECTION	0x0000038F	钢包编号识别事件(对应 DEV_EVENT_LADLE_NO_DETECTION_INFO)
EVENT_IVS_FISHING_DETECTION	0x00000390	钓鱼检测事件(对应 DEV_EVENT_FISHING_DETECTION_INFO)
EVENT_IVS_ALARM_ACCESSORY	0x00000391	报警配件事件(对应 DEV_EVENT_ALARM_ACCESSORY_INFO)
EVENT_IVS_WIRELESS_DEV_LOWPOWER	0x00000392	无线设备低电量事件(对应 DEV_EVENT_WIRELESS_DEV_LOWPOWER_INFO)
EVENT_IVS_SENSOR_ABNORMAL	0x00000393	探测器异常报警(对应 DEV_EVENT_SENSOR_ABNORMAL_INFO)
EVENT_IVS_MODULE_LOST	0x00000394	模块断线报警事件(对应 DEV_EVENT_MODULE_LOST_INFO)
EVENT_IVS_CROWD_LEVEL_DETECTION	0x00000395	拥挤程度检测事件(对应 DEV_EVENT_CROWD_LEVEL_DETECTION_INFO)
EVENT_IVS_REGION_DEFENSE_DETECTION	0x00000396	区域看防事件(对应 DEV_EVENT_REGION_DEFENSE_DETECTION_INFO)
EVENT_IVS_DUSTBIN_DETECTION	0x00000397	垃圾桶检测事件(对应 DEV_EVENT_DUSTBIN_DETECTION_INFO)
EVENT_IVS_DIALRECOGNITION_EX	0x00000398	仪表检测事件(对应 DEV_EVENT_DIALRECOGNITION_INFO)
EVENT_IVS_OCR_DETECTION	0x00000399	OCR 检测事件(对应 DEV_EVENT_OCR_DETECTION_INFO)
EVENT_IVS_ROAD_CONDITIONS_DETECTION	0x0000039A	路面检测事件(对应 DEV_EVENT_ROAD_CONDITIONS_DETECTION_INFO)
EVENT_IVS_NEAR_OBJECT_DETECT	0x0000039B	近物体检测事件(对应 DEV_EVENT_NEAR_OBJECT_DETECT_INFO)

智能事件名称	序号	含义
EVENT_IVS_OBJECT_NUM_DETECTION	0x0000039C	区域内计数事件(对应 DEV_EVENT_OBJECT_NUM_DETECTION_INFO, 注意该事件不支持单独使用，仅作为“开放智能事件(EVENT_IVS_OPEN_INTELLI)”中的一种规则)
EVENT_IVS_OPEN_INTELLI	0x0000039D	开放智能事件(对应 DEV_EVENT_OPEN_INTELLI_INFO)
EVENT_IVS_CRANE_LOAD_STAY_DETECTION	0x0000039E	天车吊物滞留事件(对应 DEV_EVENT_CRANE_LOAD_STAY_DETECTION_INFO)
EVENT_IVS_TRAFFIC_DRIVER_NO_BELT	0x00000400	未系安全带报警事件(对应 DEV_EVENT_TRAFFIC_DRIVER_NO_BELT_INFO)
EVENT_IVS_RIDING_MOTOR_CYCLE	0x00000401	摩托车骑跨检测事件(对应 DEV_EVENT_RIDING_MOTOR_CYCLE_INFO)
EVENT_IVS_CONVEYOR_ARTICLE_TYPE	0x00000402	传送带异物检测事件(对应 DEV_EVENT_CONVEYOR_ARTICLE_TYPE_INFO)
EVENT_IVS_BLIND_ALARM	0x00000403	盲区报警事件(对应 DEV_EVENT_BLIND_ALARM_INFO)
EVENT_IVS_TRAFFIC_SPEED_DROP_SHARPLY	0x00000404	车辆速度剧减事件(对应 DEV_EVENT_TRAFFIC_SPEED_DROP_SHARPLY_INFO)
EVENT_IVS_GESTURE_DETECTION	0x00000405	手势检测事件(对应 DEV_EVENT_GESTURE_DETECTION_INFO)
EVENT_IVS_HEAD_LIFT_DETECTION	0x00000406	抬头检测事件(对应 DEV_EVENT_HEAD_LIFT_DETECTION_INFO)
EVENT_IVS_DRAINING_DETECTION	0x00000407	排水检测识别事件(对应 DEV_EVENT_DRAINING_DETECTION_INFO)
EVENT_IVS_TRAFFIC_CROSSING_SPEEDY	0x00000408	斑马线不减速事件(对应 DEV_EVENT_TRAFFIC_CROSSING_SPEEDY_INFO)
EVENT_IVS_TRAFFIC_LARGECAR_NO_STOP	0x00000409	大车右转不停车事件(对应 DEV_EVENT_TRAFFIC_LARGECAR_NO_STOP_INFO)
EVENT_IVS_TRAFFIC_OVERTAKE_ONRIGHT	0x0000040A	右侧超车事件(对应 DEV_EVENT_TRAFFIC_OVERTAKE_ONRIGHT_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_TRUCK_OCCUPIED	0x0000040B	大车占道事件(对应 DEV_EVENT_TRAFFIC_TRUCK_OCCUPIED_INFO)
EVENT_IVS_TRAFFIC_SMALL_DISPLACEMENT	0x0000040C	车辆缓行事件(对应 DEV_EVENT_TRAFFIC_SMALL_DISPLACEMENT_INFO)
EVENT_IVS_GREEN_BELT	0x0000040D	绿化带报警事件(对应 DEV_EVENT_GREEN_BELT_ALARM_INFO)
EVENT_IVS_ROAD_DAMAGE	0x0000040E	道路损坏报警事件(对应 DEV_EVENT_ROAD_DAMAGE_ALARM_INFO)
EVENT_IVS_TRAFFIC_SERPENTINE_CHANGE_LANE	0x0000040F	蛇形变道事件(对应 DEV_EVENT_TRAFFIC_SERPENTINE_CHANGE_LANE_INFO)
EVENT_IVS_PERSON_TRANS_DETECTION	0x00000410	人员传输检测事件(对应 DEV_EVENT_IVS_PERSON_TRANS_DETECTION_INFO)
EVENT_IVS_HUMAN_ANIMAL_COEXISTENCE	0x00000411	人和动物检测事件(对应 DEV_EVENT_HUMAN_ANIMAL_COEXISTENCE_INFO)
EVENT_IVS_TANK_CAPACITY_DETECTION	0x00000412	贮柜储量检测事件(对应 DEV_EVENT_IVS_TANK_CAPACITY_DETECTION_INFO)
EVENT_IVS_TANK_DUMPING_DETECTION	0x00000413	周转箱倒料不干净检测事件(对应 DEV_EVENT_IVS_TANK_DUMPING_DETECTION_INFO)
EVENT_IVS_TANK_OVERFLOW_DETECTION	0x00000414	尘棒箱满溢检测事件(对应 DEV_EVENT_IVS_TANK_OVERFLOW_DETECTION_INFO)
EVENT_IVS_DUSTBIN_RETREAT	0x00000416	垃圾桶撤离检测事件(对应 NET_DEV_EVENT_DUSTBIN_RETREAT_INFO)
EVENT_IVS_SCRAPSTEEL_DETECT	0x00000417	废钢检测事件(对应 NET_DEV_EVENT_SCRAPSTEEL_DETECT_INFO)
EVENT_IVS_SIGNAL_LIGHT_ON_DETECTION	0x00000418	信号灯同亮检测事件(对应 NET_DEV_EVENT_SIGNAL_LIGHT_ON_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_SIGNAL_LIGHT_FLASHYELLOW_DETECTION	0x00000419	单色灯黄闪检测事件(对应 NET_DEV_EVENT_SIGNAL_LIGHT_FLASHYELLOW_DETECTION_INFO)
EVENT_IVS_SIGNAL_LIGHT_NOBRIGHT_DETECTION	0x0000041A	信号灯亮度不足检测事件(对应 NET_DEV_EVENT_SIGNAL_LIGHT_NOBRIGHT_DETECTION_INFO)
EVENT_IVS_SIGNAL_LIGHT_BLOCKING_DETECTION	0x0000041B	信号灯遮挡检测事件(对应 NET_DEV_EVENT_SIGNAL_LIGHT_BLOCKING_DETECTION_INFO)
EVENT_IVS_SIGNAL_LIGHT_ALWAYSON_DETECTION	0x0000041C	单色灯常亮检测事件(对应 NET_DEV_EVENT_SIGNAL_LIGHT_ALWAYSON_DETECTION_INFO)
EVENT_IVS_LEDSCREEN_NOTLIGHTUP_DETECTION	0x0000041D	LED 屏不亮检测事件(对应 NET_DEV_EVENT_LEDSCREEN_NOTLIGHTUP_DETECTION_INFO)
EVENT_IVS_SIGNAL_LIGHTOFF_DETECTION	0x0000041E	信号灯不亮检测事件(对应 NET_DEV_EVENT_SIGNAL_LIGHTOFF_DETECTION_INFO)
EVENT_IVS_PIC_COMPARE_DETECTION	0x0000041F	图片比对不上报警事件(对应 NET_DEV_EVENT_PIC_COMPARE_DETECTION_INFO)
EVENT_IVS_CHANGE_DETECTION	0x00000420	差异检测事件(对应 NET_DEV_EVENT_CHANGE_DETECTION_INFO)
EVENT_IVS_LINK_AGE_RADAR_ALARM	0x00000421	联动事件(作为细节事件、联动球机时发送)(对应 NET_DEV_EVENT_LINK_AGE_RADAR_ALARM_INFO)
EVENT_IVS_VEHICLE_STATE	0x00000422	车辆状态事件(对应 NET_DEV_EVENT_VEHICLE_STATE_INFO)
EVENT_IVS_PASS_WINDOW_DELIVERY	0x00000423	隔窗递物检测事件(对应 NET_DEV_EVENT_PASS_WINDOW_DELIVERY_INFO)
EVENT_IVS_DOOR_STATE_DETECTION	0x00000424	开关门检测事件(对应 NET_DEV_EVENT_DOOR_STATE_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_WASTE_MIXED_INVEST	0x00000425	垃圾混投事件(对应 NET_DEV_EVENT_WASTE_MIXED_INVEST_INFO)
EVENT_IVS_UNBROKEN_TRASHBAG	0x00000426	垃圾袋未破袋检测事件(对应 NET_DEV_EVENT_UNBROKEN_TRASHBAG_INFO)
EVENT_IVS_PERSON_CARRY_TRASHBAG	0x00000427	人员拎袋报警事件(对应 NET_DEV_EVENT_PERSON_CARRY_TRASHBAG_INFO)
EVENT_IVS_UNATTENDED_DETENTION	0x00000428	无人值守检测事件(对应 NET_DEV_EVENT_UNATTENDED_DETENTION_INFO)
EVENT_IVS_DROP_DETECTION	0x00000429	滴漏检测事件(对应 NET_DEV_EVENT_DROP_DETECTION_INFO)
EVENT_IVS_TEMPERATURE_ALARM	0x0000042A	温度报警事件(对应 NET_DEV_EVENT_TEMPERATURE_ALARM_INFO)
EVENT_IVS_HUMIDITY_ALARM	0x0000042B	湿度报警事件(对应 NET_DEV_EVENT_HUMIDITY_ALARM_INFO)
EVENT_IVS_POWER_SWITCHER_ALARM	0x0000042C	电量报警事件(对应 NET_DEV_EVENT_POWER_SWITCHER_ALARM_INFO)
EVENT_IVS_GAS_TANK_DETECTION	0x0000042D	煤气罐检测事件(对应 NET_DEV_EVENT_GAS_TANK_DETECTION_INFO)
EVENT_IVS_TRAFFIC_OCCUPYING_THEPATH	0x0000042E	占道事件(对应 NET_EVENT_TRAFFIC_OCCUPYING_THEPATH_INFO)
EVENT_IVS_ILLEGAL_CARRIAGE	0x0000042F	非法运输事件(对应 NET_DEV_EVENT_ILLEGAL_CARRIAGE_INFO)
EVENT_IVS_SILICON_FIREPUTTER_DETECTION	0x00000430	刺火推杆检测事件(对应 NET_EVENT_SILICON_FIREPUTTER_DETECTION_INFO)
EVENT_IVS_ALARM_METHANE_ALARM	0x00000431	甲烷报警事件(对应结构体 NET_DEV_EVENT_METHANE_ALARM_INFO)

智能事件名称	序号	含义
EVENT_IVS_ALARM_TORPEDO_DETECT	0x00000432	鱼雷罐报警事件(对应结构体 NET_DEV_EVENT_TORPEDO_DETECT_ALARM_INFO)
EVENT_IVS_FINANCIAL_CABINET_ALARM_EVENT	0x00000433	金融业务库异常事件(对应 NET_DEV_EVENT_FINANCIAL_CABINET_ALARM_EVENT_INFO)
EVENT_IVS_SCRAPSTEEL_DANGER_DETECT	0x00000434	废钢危险品事件(对应 NET_DEV_EVENT_SCRAPSTEEL_DANGER_DETECT_INFO)
EVENT_IVS_TRAFFIC_RUNASTOP_SIGN	0x00000435	闯行停车标志事件(对应 NET_DEV_EVENT_TRAFFIC_RUNASTOP_SIGN_INFO)
EVENT_IVS_IRCUT_MODESWITCH_ALARM_EVENT	0x00000436	IRCUT 模式切换事件(对应 NET_DEV_EVENT_IRCUT_MODESWITCH_ALARM_INFO)
EVENT_IVS_TRAFFIC_VEHICLE_OVERLOAD	0x00000437	超员检测事件(对应 NET_DEV_EVENT_TRAFFIC_VEHICLE_OVERLOAD_INFO)
EVENT_IVS_REMOTE_APPROVAL_ALARM	0x00000438	金融远程审批事件(对应 NET_DEV_EVENT_REMOTE_APPROVAL_ALARM_INFO)
EVENT_IVS_ANTI_COUNTERFEIT	0x00000439	防造假检测事件(对应 NET_DEV_EVENT_ANTI_COUNTERFEIT_INFO)
EVENT_IVS_TRAFFIC_SPEED_DIFFPREWARNING	0x0000043A	速度差预警事件(对应 NET_DEV_EVENT_TRAFFIC_SPEED_DIFFPREWARNING_INFO)
EVENT_IVS_TRAFFIC_DRIVER_IDENTIFIED	0x0000043B	驾驶员比对成功通知事件(对应 NET_DEV_EVENT_TRAFFIC_DRIVER_IDENTIFIED_INFO)
EVENT_IVS_TRAFFIC_QUEUE_OVERFLOW	0x0000043C	排队溢出事件(对应 NET_DEV_EVENT_TRAFFIC_QUEUE_OVERFLOW_INFO)
EVENT_IVS_TRAFFIC_QUEUE_TIMEOUT	0x0000043D	排队超限事件(对应 NET_DEV_EVENT_TRAFFIC_QUEUE_TIMEOUT_INFO)

智能事件名称	序号	含义
EVENT_IVS_RAILING_PASS_DETECTION	0x0000043E	隔栏传物事件(对应结构体 NET_DEV_EVENT_RAILING_PASS_DETECTION_INFO)
EVENT_IVS_MULTI_MAN_NUM_DETECTION	0x0000043F	讯问会见室人数报警事件(对应 NET_DEV_EVENT_MULTI_MAN_NUM_DETECTION_INFO)
EVENT_IVS_OBJECT_QUANTITY_DETECTION	0x00000440	目标类型和数量检测报警事件(对应 NET_DEV_EVENT_OBJECT_QUANTITY_DETECTION_INFO)
EVENT_IVS_USERMANAGER_FOR_TWSDK	0x00000441	用户信息上报事件(对应 NET_DEV_EVENT_USERMANAGER_FOR_TWSDK_INFO)
EVENT_IVS_DRIVE_ASSISTANT_ALARM	0x00000442	驱动辅助报警(对应 NET_DEV_EVENT_DRIVE_ASSISTANT_ALARM_INFO)
EVENT_IVS_CROSSLINE_STAT	0x00000443	绊线计数事件(对应 NET_DEV_EVENT_CROSSLINE_STAT_INFO),(注意该事件暂时不支持单独使用，仅作为“开放智能事件(EVENT_IVS_OPEN_INTELLI)”中的一种规则))
EVENT_IVS_REGIONNUM_STAT	0x00000444	域内计数事件(对应 NET_DEV_EVENT_REGIONNUM_STAT_INFO),(注意该事件暂时不支持单独使用，仅作为“开放智能事件(EVENT_IVS_OPEN_INTELLI)”中的一种规则))
EVENT_IVS_FISH_STATE_DETECTION	0x00000445	鱼类状态检测报警(对应 NET_DEV_EVENT_FISH_STATE_DETECTION_INFO)
EVENT_IVS_CONVEYOR_NTH_TO_STH_DETECTION	0x00000446	传送带载体从无到有检测报警事件(对应 NET_DEV_EVENT_CONVEYOR_NTH_TO_STH_DETECTION_INFO)
EVENT_IVS_POSITION_SNAP	0x00000447	按位置抓图事件(对应 NET_DEV_EVENT_POSITION_SNAP_INFO)
EVENT_IVS_DRIVER_MISMATCH_CERTIFICATE	0x00000448	司机人证不符事件(对应 NET_DEV_EVENT_DRIVER_MISMATCH_CERTIFICATE_INFO)

智能事件名称	序号	含义
EVENT_IVS_DRIVER_MISMATCH_VEHICLE	0x00000449	司机人车不符事件(对应 NET_DEV_EVENT_DRIVER_MISMATCH_VEHICLE_INFO)
EVENT_IVS_CERTIFICATE_MISMATCH_VEHICLE	0x0000044A	司机证车不符事件(对应 NET_DEV_EVENT_CERTIFICATE_MISMATCH_VEHICLE_INFO)
EVENI_IVS_XRAY_DETECT_PACKAGE	0x0000044B	X 光机检测包裹事件(对应 NET_EVENT_XRAY_DETECT_PACKAGE_INFO)
EVENT_IVS_CONVEYOR_STH_TO_NTH_DETECTION	0x0000044C	传送带载体从有到无检测报警事件(对应 NET_DEV_EVENT_CONVEYOR_STH_TO_NTH_DETECTION_INFO)
EVENT_IVS_CO_LOW_ALARM	0x0000044D	CO 低浓度报警事件(对应 NET_DEV_EVENT_CO_LOW_ALARM_INFO)
EVENT_IVS_CO_HIGH_ALARM	0x0000044E	CO 高浓度报警事件(对应 NET_DEV_EVENT_CO_HIGH_ALARM_INFO)
EVENT_IVS_TIMECHANGE_FOR_TWSDK	0x0000044F	系统时间被修改报警事件(对应 NET_DEV_EVENT_TIMECHANGE_FOR_TWSDK_INFO)
EVENT_IVS_CIGARETTE_CASE_DETECTION	0x00000450	烟盒检测事件(对应 NET_DEV_EVENT_CIGARETTE_CASE_DETECTION_INFO)
EVENT_IVS_CONVEYOR_BELT_STATUS	0x00000451	传送带运动状态检测报警事件(对应 NET_DEV_EVENT_CONVEYOR_BELT_STATUS_INFO)
EVENT_IVS_HOSPITAL_TALK_CONTROL	0x00000452	设备上报谈话事件(对应 NET_DEV_EVENT_HOSPITAL_TALK_CONTROL_INFO)
EVENT_IVS_PASSENGER_FLOW_ALARM	0x00000453	校车上下客事件(对应 NET_DEV_EVENT_PASSENGER_FLOW_ALARM_INFO)
EVENT_IVS_SCHOOL_BUS_SWIPE_CARD	0x00000454	学生刷卡事件(对应 NET_DEV_EVENT_SCHOOL_BUS_SWIPE_CARD_INFO)
EVENT_IVS_COLD_SPOT_WARNING	0x00000455	冷点报警(对应 NET_DEV_EVENT_COLD_SPOT_WARNING_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_PLATE_ABNORMAL	0x00000456	车牌异常检测(对应 NET_DEV_EVENT_TRAFFIC_PLATE_ABNORMAL_INFO)
EVENT_IVS_TRAFFIC_ACCELERATION_RAPID	0x00000457	急加速事件(对应 NET_DEV_EVENT_TRAFFIC_ACCELERATION_RAPID_INFO)
EVENT_IVS_TRAFFIC_TURN_SHARP	0x00000458	急转弯事件(对应 NET_DEV_EVENT_TRAFFIC_TURN_SHARP_INFO)
EVENT_IVS_GARBAGE_PLASTICBAG	0x00000459	打包垃圾检测事件(对应 NET_DEV_EVENT_GARBAGE_PLASTICBAG_INFO)
EVENT_IVS_DOOR_STATUS_FOR_BOX	0x0000045A	智能柜门状态上报(对应 NET_DEV_EVENT_DOOR_STATUS_FOR_BOX_INFO)
EVENT_IVS_COLLISION_CONFLICT	0x0000045B	碰撞冲突事件(对应 NET_DEV_EVENT_COLLISION_CONFLICT_INFO)
EVENT_IVS_PHOTOGRAPH_DETECTION	0x0000045C	拍摄行为检测事件(对应 NET_DEV_EVENT_PHOTOGRAPH_DETECTION_INFO)
EVENT_IVS_REFUELING_GUN_PERSON	0x0000045D	提枪识别动作事件(对应 NET_DEV_EVENT_REFUELING_GUN_PERSON_INFO)
EVENT_IVS_AUDIO_MUTATION	0x0000045E	声强突变事件(对应 NET_DEV_EVENT_AUDIO_MUTATION_INFO),(注意该事件暂时不支持单独使用，仅作为“开放智能事件(EVENT_IVS_OPEN_INTELLI)”中的一种规则))
EVENT_IVS_OBJECT_APPEAR_DETECTION	0x0000045F	目标出现事件(对应 NET_DEV_EVENT_OBJECT_APPEAR_DETECTION_INFO)
EVENT_IVS_OBJECT_DISAPPEAR_DETECTION	0x00000460	目标消失事件(对应 NET_DEV_EVENT_OBJECT_DISAPPEAR_DETECTION_INFO)
EVENT_IVS_OBJECT_STATE_DETECTION	0x00000461	目标状态事件(对应 NET_DEV_EVENT_OBJECT_STATE_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAPPED_IN_LIFT_DETECTION	0x00000462	电梯困人检测(对应 NET_DEV_EVENT_TRAPPED_IN_LIFT_DETECTION_INFO)
EVENT_IVS_LIGHT_DETECTION	0x00000463	视频异常事件(对应 NET_DEV_EVENT_LIGHT_DETECTION_INFO)
EVENT_IVS_COVERING_DETECTION	0x00000464	遮挡物检测事件(对应 NET_DEV_EVENT_COVERING_DETECTION_INFO)
EVENT_IVS_ATM_SMASH_DETECTION	0x00000465	ATM 打砸检测事件(对应 NET_DEV_EVENT_ATM_SMASH_DETECTION_INFO)
EVENT_IVS_TRAFFIC_ASSISTANT_DRIVER_SMOKING	0x00000466	副驾驶抽烟事件(对应 NET_DEV_EVENT_TRAFFIC_ASSISTANT_DRIVER_SMOKING_INFO)
EVENT_IVS_TRAFFIC_ASSISTANT_DRIVER_CALLING	0x00000467	副驾驶打电话事件(对应 NET_DEV_EVENT_TRAFFIC_ASSISTANT_DRIVER_CALLING_INFO)
EVENT_IVS_TRAFFIC_VEHICLEIN_AREA	0x00000468	有车禁入事件(对应 NET_DEV_EVENT_TRAFFIC_VEHICLEIN_AREA_INFO)
EVENT_IVS_TRAFFIC_NOT_AFTER_HORIZONTAL_PEOPLE	0x00000469	不礼让行人事件(对应 NET_DEV_EVENT_TRAFFIC_NOT_AFTER_HORIZONTAL_PEOPLE_INFO)
EVENT_IVS_WATER_LOGGED_DETECTION	0x0000046A	内涝积水事件(对应 NET_DEV_EVENT_WATER_LOGGED_DETECTION_INFO)
EVENT_IVS_TRAFFIC_DRIVER_SUN_VISOR	0x0000046B	驾驶员遮阳板检测事件(对应 NET_DEV_EVENT_TRAFFIC_DRIVER_SUN_VISOR_INFO)
EVENT_IVS_WORK_CHECK	0x0000046C	自动查岗事件(对应 NET_DEV_EVENT_WORK_CHECK_INFO)
EVENT_IVS_PERSON_PECCANCY	0x0000046D	人员违章事件(对应 NET_DEV_EVENT_PERSON_PECCANCY_INFO)
EVENT_IVS_ACTION_COUNT	0x0000046E	行为自定义行为计数事件(对应 NET_DEV_EVENT_ACTION_COUNT_INFO)

智能事件名称	序号	含义
EVENT_IVS_WADING_DETECTION	0x0000046F	涉水安全检测、水域监测报警(对应 NET_DEV_EVENT_WADING_DETECTION_INFO)
EVENT_IVS_TRAFFIC_GO_STRAIGHT	0x00000470	违章直行违法事件(对应 NET_DEV_EVENT_TRAFFIC_GO_STRAIGHT_INFO)
EVENT_IVS_PHONE_SECURITY_STICKER_DETECTION	0x00000471	手机保密贴检测事件(对应 NET_DEV_EVENT_PHONE_SECURITY_STICKER_DETECTION_INFO)
EVENT_IVS_SAME_OBJECT_SEARCH_DETECT	0x00000472	按图索骥物品检测事件(对应 NET_DEV_EVENT_SAME_OBJECT_SEARCH_DETECT_INFO)
EVENT_IVS_TRAFFIC_PEOPLE_IN_MOTOR_ROUTE	0x00000473	行人使用机动车道事件(对应 NET_DEV_EVENT_TRAFFIC_PEOPLE_IN_MOTOR_ROUTE_INFO)
EVENT_IVS_TRAFFIC_PEOPLE_IN_NON_MOTOR_ROUTE	0x00000474	行人使用非机车道事件(对应 NET_DEV_EVENT_TRAFFIC_PEOPLE_IN_NON_MOTOR_ROUTE_INFO)
EVENT_IVS_ALL_INTELLIGENCE	0x00000475	所有智能事件(并非下发所有智能事件，对应智能录像查询)
EVENT_IVS_GATA_HEIGHT_DETECTION	0x00000476	水闸门高度事件(对应 NET_DEV_EVENT_GATA_HEIGHT_DETECTION_INFO)
EVENT_IVS_TRAFFIC_BRIGHT_LIGHT	0x00000477	非法加装牛眼灯事件(对应 NET_DEV_EVENT_TRAFFIC_BRIGHT_LIGHT_INFO)
EVENT_IVS_FREQUENT_CHANGE_LANE	0x00000478	高级驾驶辅助频繁变道报警(对应 NET_DEV_EVENT_FREQUENT_CHANGE_LANE_INFO)
EVENT_IVS_PYROTECHNIC_DETECTION	0x00000479	烟火监测报警(对应 NET_DEV_EVENT_PYROTECHNIC_DETECTION_INFO)
EVENT_IVS_TRAFFIC_STRANGE_CAR	0x0000047A	陌生车辆事件(对应 NET_DEV_EVENT_TRAFFIC_STRANGE_CAR_INFO)

智能事件名称	序号	含义
EVENT_IVS_SAME_OBJECT_SEARCH_COUNT	0x00000480	按图索骥物品计数事件(对应 NET_DEV_EVENT_SAME_OBJECT_SEARCH_COUNT_INFO)
EVENT_IVS_VIDEO_OUT_SNAP	0x00000481	输出口抓屏事件(对应 NET_DEV_EVENT_IVS_VIDEO_OUT_SNAP_INFO)
EVENT_IVS_POSE_EXCEPTION_NUM_DETECTION	0x00000482	特殊姿态的目标数量异常检测报警事件(对应 NET_DEV_EVENT_IVS_POSE_EXCEPTION_NUM_DETECTION_INFO)
EVENT_IVS_HOSPITAL_APPRAISE	0x00000483	医护设备上报评价事件(对应 NET_DEV_EVENT_HOSPITAL_APPRAISE_INFO)
EVENT_IVS_GRANARY_TRANS_ACTION_DETECTION	0x00000484	粮面异动检测事件上报(对应 NET_DEV_EVENT_GRANARY_TRANS_ACTION_DETECTION_INFO)
EVENT_IVS_REGION_PROPORTION_DETECTION	0x00000485	区域占比检测事件(对应 NET_DEV_EVENT_REGION_PROPORTION_DETECTION_INFO)
EVENT_IVS_TRAFFIC_DANGEROUS_SPEED	0x00000486	危险超速事件(对应 NET_DEV_EVENT_TRAFFIC_DANGEROUS_SPEED_INFO)
EVENT_IVS_TRAFFIC_URUN_RED_LIGHT	0x00000487	掉头闯红灯事件(对应 NET_DEV_EVENT_TRAFFIC_URUN_RED_LIGHT_INFO)
EVENT_IVS_TRAFFIC_LEFT_WRONG_ROUTE	0x00000488	不按导向箭头左行事件(对应 NET_DEV_EVENT_TRAFFIC_LEFT_WRONG_ROUTE_INFO)
EVENT_IVS_TRAFFIC_RIGHT_WRONG_ROUTE	0x00000489	不按导向箭头右行事件(对应 NET_DEV_EVENT_TRAFFIC_RIGHT_WRONG_ROUTE_INFO)
EVENT_IVS_TRAFFIC_STRAIGHT_WRONG_ROUTE	0x0000048A	不按导向箭头直行事件(对应 NET_DEV_EVENT_TRAFFIC_STRAIGHT_WRONG_ROUTE_INFO)
EVENT_IVS_TRAFFIC_UWRONG_ROUTE	0x0000048B	不按导向箭头掉头事件(对应 NET_DEV_EVENT_TRAFFIC_UWRONG_ROUTE_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_UNQUEUED	0x0000048C	未排队等待事件(对应 NET_DEV_EVENT_TRAFFIC_UNQUEUED_INFO)
EVENT_IVS_POWER_LINE_CHANGE	0x0000048D	线路舞动带图事件(对应 NET_DEV_EVENT_POWER_LINE_CHANGE_INFO)
EVENT_IVS_ILLEGAL_ADBOARD_DETECTION	0x0000048E	非法广告牌检测报警(对应 NET_DEV_EVENT_ILLEGAL_ADBOARD_DETECTION_INFO)
EVENT_IVS_GRAIN_HEIGHT_DETECTION	0x0000048F	动粮检测事件(对应 NET_DEV_EVENT_GRAIN_HEIGHT_DETECTION_INFO)
EVENT_IVS_OBJECTCHANGE_DETECTION	0x00000490	变化事件目标变化检测(对应 NET_DEV_EVENT_OBJECTCHANGE_DETECTION_INFO)
EVENT_IVS_FIRE_AWAY_FROM_MAN_DETECTION	0x00000491	动火离人检测(对应 NET_DEV_EVENT_FIRE_AWAY_FROM_MAN_DETECTION_INFO)
EVENT_IVS_TRUCK_NOT_CLEAN_FOR_PRMA	0x00000492	工程车未清洗事件(对应 NET_DEV_EVENT_TRUCK_NOT_CLEAN_FOR_PRMA_INFO)
EVENT_IVS_ELEVATOR_WORK_INFO	0x00000493	上报电梯运行数据事件(对应 NET_DEV_EVENT_ELEVATOR_WORK_INFO_INFO)
EVENT_IVS_ELEVATOR_ALARM	0x00000494	电梯异常报警(对应 NET_DEV_EVENT_ELEVATOR_ALARM_INFO)
EVENT_IVS_MATERIAL_FLOW_STATUS_DETECTION	0x00000495	料流状态识别检测 (对应 NET_DEV_EVENT_MATERIAL_FLOW_STATUS_DETECTION_INFO)
EVENT_IVS_RULE_CORRELATION_ALARM	0x00000496	行为组合事件(对应 NET_DEV_EVENT_RULE_CORRELATION_ALARM_INFO)
EVENT_IVS_TRAFFIC_OVER_PASSERSBY_LINE	0x00000497	侵占人行横道线事件(对应 NET_DEV_EVENT_OVER_PASSERSBY_LINE_INFO)

智能事件名称	序号	含义
EVENT_IVS_TRAFFIC_OVER_NONE_MOTOR_LINE	0x00000498	侵占二轮车横道线事件(对应 NET_DEV_EVENT_OVER_NONE_MOTOR_LINE_INFO)
EVENT_IVS_TRAFFIC_TRUST_CAR	0x00000499	信任车辆事件(对应 NET_DEV_EVENT_TRAFFIC_TRUST_CAR_INFO)
EVENT_IVS_TRAFFIC_SUSPICIOUS_CAR	0x0000049A	嫌疑车辆事件(对应 NET_DEV_EVENT_TRAFFIC_SUSPICIOUS_CAR_INFO)
EVENT_IVS_TRAFFIC_RAIN_DETECTION	0x0000049B	雨天检测(对应 NET_DEV_EVENT_TRAFFIC_RAIN_DETECTION_INFO)
EVENT_IVS_TRAFFIC_SNOW_DETECTION	0x0000049C	雪天检测(对应 NET_DEV_EVENT_TRAFFIC_SNOW_DETECTION_INFO)
EVENT_IVS_TRAFFIC_FOG_DETECTION	0x0000049D	雾天检测(对应 NET_DEV_EVENT_TRAFFIC_FOG_DETECTION_INFO)
EVENT_IVS_CROP_DETECTION	0x0000049E	农作物检测事件(对应 NET_DEV_EVENT_CROP_DETECTION_INFO)
EVENT_IVS_ENCLOSURE_ALARM	0x0000049F	离站数据事件(对应 NET_DEV_EVENT_ENCLOSURE_ALARM_INFO)
EVENT_IVS_HEAT_IMAGING_TEMPER_INFO	0x000004A0	热成像测温点温度信息异常报警(对应结构体 DEV_EVENT_HEAT_IMAGING_TEMPER_INFO_DETAIL)
EVENT_IVS_TRAFFIC_STRAIGHTRUN_REDLIGHT	0x000004A1	直行闯红灯事件(对应 NET_DEV_EVENT_TRAFFIC_STRAIGHTRUN_REDLIGHT_INFO)
EVENT_IVS_TRAFFIC_LEFTRUN_REDLIGHT	0x000004A2	左转闯红灯(对应 NET_DEV_EVENT_TRAFFIC_LEFTRUN_REDLIGHT_INFO)
EVENT_IVS_TRAFFIC_RIGHTRUN_REDLIGHT	0x000004A3	右转闯红灯(对应 NET_DEV_EVENT_TRAFFIC_RIGHTRUN_REDLIGHT_INFO)

智能事件名称	序号	含义
EVENT_IVS_GROUND_THING_DETECTION	0x000004A4	地物识别(对应 NET_DEV_EVENT_GROUND_THING_DETECTION_INFO)
EVENT_IVS_PERSON_SPACE_TRACK	0x000004A5	人员空间轨迹(对应 NET_DEV_EVENT_PERSON_SPACE_TRACK_INFO)
EVENT_IVS_AUDIO_SAMPLE	0x000004A6	音频数据定时采集上报事件(对应 NET_DEV_EVENT_AUDIO_SAMPLE_INFO),(注意该事件暂时不支持单独使用，仅作为“开放智能事件(EVENT_IVS_OPEN_INTELLI)”中的一种规则))
EVENT_IVS_RADARANALYSE_DETECTION	0x000004A7	雷达联动事件(对应 NET_DEV_EVENT_RADARANALYSE_DETECTION_INFO)
EVENT_IVS_LADLE_HANGING_DETECTION	0x000004A8	挂耳检测事件(对应 NET_DEV_EVENT_LADLE_HANGING_DETECTION_INFO)
EVENT_IVS_TRAFFIC_NONMOTOR_CALLING	0x000004A9	非机动车打电话(对应 NET_DEV_EVENT_TRAFFIC_NONMOTOR_CALLING_INFO)
EVENT_IVS_TRAFFIC_NONMOTOR_IN_ZEBRA_CROSSING	0x000004AA	非机动车占用斑马线(对应 NET_DEV_EVENT_TRAFFIC_NONMOTOR_IN_ZEBRA_CROSSING_INFO)
EVENT_IVS_TRAFFIC_TURN_RIGHT_AFTER_PEOPLE_SUCCESS	0x000004AB	右转成功礼让直行行人(对应 NET_DEV_EVENT_TRAFFIC_TURN_RIGHT_AFTER_PEOPLE_SUCCESS_INFO)
EVENT_IVS_PSR_STAND_RISE_DETECTION	0x000004AC	qf 站立起身事件(对应 NET_DEV_EVENT_PSR_STAND_RISE_DETECTION_INFO)
EVENT_IVS_PSR_KEY_PEO_RISE_DETECTION	0x000004AD	重点人员起身事件(对应 NET_DEV_EVENT_PSR_KEY_PEO_RISE_DETECTION_INFO)
EVENT_IVS_NATURAL_DISASTER_DETECTION	0x000004AE	自然灾害检测(对应 NET_DEV_EVENT_NATURAL_DISASTER_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_PRE_HEAT_IMAGING_TEMPER	0x000004AF	热成像测温点温度预报警(对应结构体 DEV_EVENT_PRE_HEAT_IMAGING_TEMPER_INFO)
EVENT_IVS_ILLEGAL_CYCLING_DETECTION	0x000004B0	非法骑行检测事件(对应结构体 NET_DEV_EVENT_ILLEGAL_CYCLING_DETECTION_INFO)
EVENT_IVS_TRAFFIC_INTERSECTION_CONGESTION	0x000004B1	路口打结事件(对应结构体 NET_DEV_EVENT_TRAFFIC_INTERSECTION_CONGESTION_INFO)
EVENT_IVS_MOSAIC	0x000004B2	隐私打码(对应结构体 NET_DEV_EVENT_MOSAIC_INFO)
EVENT_IVS_FLAME_EXTINGUISH	0x000004B3	火焰熄灭事件(对应结构体 NET_DEV_EVENT_FLAME_EXTINGUISH_INFO)
EVENT_IVS_FLAME_OVER_SIZE	0x000004B4	火焰过大事件(对应结构体 NET_DEV_EVENT_FLAME_OVER_SIZE_INFO)
EVENT_IVS_FLAME_UNDER_SIZE	0x000004B5	火焰过小事件(对应结构体 NET_DEV_EVENT_FLAME_UNDER_SIZE_INFO)
EVENT_IVS_ATTENDANCE_RATE_DETECTION	0x000004B6	到课率检测事件(对应结构体 NET_DEV_EVENT_ATTENDANCE_RATE_DETECTION_INFO)
EVENT_IVS_SEATING_RATE_DETECTION	0x000004B7	就座率检测事件(对应结构体 NET_DEV_EVENT_SEATING_RATE_DETECTION_INFO)
EVENT_IVS_AUDIO_ANALYSIS	0x000004B8	音频语义(关键字)识别报警(对应结构体 NET_DEV_EVENT_AUDIO_ANALYSIS_INFO)
EVENT_IVS_STEAL_OIL	0x000004B9	低油量报警事件(对应结构体 NET_DEV_EVENT_STEAL_OIL_INFO)
EVENT_IVS_VISION_LANG_MODEL_DETECTION	0x000004BA	视觉文本模型检测事件(对应结构体 NET_DEV_EVENT_VISION_LANG_MODEL_DETECTION_INFO)
EVENT_IVS_BETWEEN_RULES_TEMPERDIFF_ALARM	0x000004BB	温差报警事件 (NET_DEV_EVENT_BETWEEN_RULES_TEMPERDIFF_ALARM_INFO)
EVENT_IVS_GAS_CLOUD_DETECTION	0x000004BC	气云检测事件 (NET_DEV_EVENT_GAS_CLOUD_DETECTION_INFO)

智能事件名称	序号	含义
EVENT_IVS_INFUSION_DETECTION	0x000004BD	输液检测事件(对应结构体 NET_DEV_EVENT_INFUSION_DETECTION_INFO)
EVENT_IVS_PERSONNEL_CATEGORY_COUNT	0x000004BE	人员类型统计事件(对应结构体 NET_DEV_EVENT_PERSONNEL_CATEGORY_COUNT_INFO)
EVENT_IVS_SMART_MOTION_EQUIPMEN	0x000004BF	智能动检事件(对应结构体 NET_DEV_EVENT_SMART_MOTION_EQUIPMENT_INFO)
EVENT_IVS_TARFFIC_NOT_IN_ALLOW_LIST	0x000004C0	不在白名单事件(对应结构体 NET_DEV_EVENT_TARFFIC_NOT_IN_ALLOW_LIST_INFO)
EVENT_IVS_STEREO_TAIL_DETECTION	0x000004C1	ATM 舱内尾随事件(仅增加枚举值，不在 CLIENT_RealLoadPictureEx 实现)
EVENT_IVS_TARGET_PROPORTION	0x000004C2	语义分割事件(对应结构体 NET_DEV_EVENT_TARGET_PROPORTION_INFO)
EVENT_IVS_OPEN_ALG_ARRANGE	0x000004C3	开放算法编排事件(对应结构体 NET_DEV_EVENT_OPEN_ALG_ARRANGE_INFO)
EVENT_IVS_AMW_PERSON_DETECTION	0x000004C4	毫米波安检门人体检测对外事件，预先占用事件枚举，暂不进行事件具体实现，暂无对应结构体
EVENT_IVS_SIGNAL_LIGHT_SKEW_DETECTION	0x000004C5	信号灯偏移检测事件(对应结构体 NET_DEV_EVENT_SIGNAL_LIGHT_SKEW_DETECTION_INFO)
EVENT_IVS_PUBLIC_SECURITY_ABNORMAL	0x000004C6	公共安全异常行为检测事件(对应结构体 NET_DEV_EVENT_PUBLIC_SECURITY_ABNORMAL_INFO)
EVENT_IVS_PATROL_TASK_FAILURE	0x000004C7	巡检任务下发异常事件(对应结构体 NET_DEV_EVENT_PATROL_TASK_FAILURE_INFO)
EVENT_IVS_TRAFFIC_FACILITIES_ABNORMAL	0x000004C8	交安设施异常检测事件(对应结构体 NET_DEV_EVENT_TRAFFIC_FACILITIES_ABNORMAL_INFO)

智能事件名称	序号	含义
EVENT_HY_FIRE_DETECTION	0x01000001	消防类火焰检测事件(对应 DEV_EVENT_HY_FIRE_DETECTION_INFO)
EVENT_HY_SMOG_DETECTION	0x01000002	消防类烟雾检测事件(对应 DEV_EVENT_HY_SMOG_DETECTION_INFO)
EVENT_HY_INFIRE_PASSAGE_DETECTION	0x01000003	消防类疏散通道阻塞事件(对应 DEV_EVENT_HY_INFIRE_PASSAGE_DETECTION_INFO)
EVENT_HY_OUTFIRE_PASSAGE_DETECTION	0x01000004	消防类消防车通道占用/消防登高面占用事件(对应 DEV_EVENT_HY_OUTFIRE_PASSAGE_DETECTION_INFO)
EVENT_HY_MAN_LEAVING_DETECTION	0x01000005	消防类消控室人员离岗事件(对应 DEV_EVENT_HY_MAN_LEAVING_DETECTION_INFO)

附录2 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3 网络安全建议

保障设备基本网络安全的必须措施：

1. 使用复杂密码

请参考如下建议进行密码设置：

- 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。
- 不包含账户名称或账户名称的倒序。
- 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. 及时更新固件和客户端软件

- 按科技行业的标准作业规范，设备（如 NVR、DVR 和 IP 摄像机等）的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。设备接入公网情况下，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。
- 建议您下载和使用最新版本客户端软件。

增强设备网络安全的建议措施：

1. 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、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）等物理接触行为。

2. 定期修改密码

建议您定期修改密码，以降低被猜测或破解的风险。

3. 及时设置、更新密码重置信息

设备支持密码重置功能，为了降低该功能被攻击者利用的风险，请您及时设置密码重置相关信息，包含预留手机号/邮箱、密保问题，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。

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。在攻击者多次密码尝试失败后，其对应账户及源 IP 将会被锁定。

5. 更改 HTTP 及其他服务默认端口

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，以减小被攻击者猜测服务端口的风险。

6. 使能 HTTPS

建议您开启 HTTPS，通过安全的通道访问 Web 服务。

7. MAC 地址绑定

建议您在设备端将其网关设备的 IP 与 MAC 地址进行绑定，以降低 ARP 欺骗风险。

8. 合理分配账户及权限

根据业务和管理需要，合理新增用户，并合理为其分配最小权限集合。

9. 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 SNMP、SMTP、UPnP 等功能，以降低设备面临的风险。

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，并设置复杂的加密密码和鉴权密码。
- SMTP：选择 TLS 方式接入邮箱服务器。
- FTP：选择 SFTP，并设置复杂密码。
- AP 热点：选择 WPA2-PSK 加密模式，并设置复杂密码。

10. 音视频加密传输

如果您的音视频数据包含重要或敏感内容，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取的风险。

11. 安全审计

- 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。
- 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。

12. 网络日志

由于设备存储容量限制，日志存储能力有限，如果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。

13. 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的服务。

- 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。
- 开启设备 IP/MAC 地址过滤功能，限制允许访问设备的主机范围。