



ISUPSDK（通用）

开发指南

法律声明

责任声明

在法律允许的最大范围内，本手册是“按照现状”提供，可能存在瑕疵或错误。本公司不对本手册提供任何形式的明示或默示保证，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；亦不对使用或是分发本手册导致的任何特殊、附带、偶然或间接的损害进行赔偿，包括但不限于商业利润损失、系统故障、数据或文档丢失产生的损失。

目录

第 1 章 概览	1
1.1 简介	1
1.2 系统要求	1
1.3 功能与组件	1
1.3.1 功能模块	1
1.3.2 服务组件	2
1.3.3 功能与服务的关系	4
1.4 更新历史	5
1.5 注意事项	9
第 2 章 设备注册	10
2.1 设备注册消息处理流程	10
第 3 章 预览	14
第 4 章 文件查找	24
第 5 章 回放	29
第 6 章 语音对讲	41
6.1 开启语音对讲	41
6.2 语音转发	45
6.3 通过流媒体服务开启语音对讲	47
第 7 章 云台控制	56
第 8 章 报警或事件配置	57
第 9 章 开启监听服务用于接收报警	61
第 10 章 设备维护	67
第 11 章 存储管理	68
11.1 开启存储服务	69
11.2 上传文件至存储服务	74

11.3 从存储服务下载文件	77
第 12 章 参数配置	82
12.1 配置功耗模式	84
第 13 章 通过传输文本协议进行集成	87
第 14 章 接口参考	89
14.1 注册模块	89
14.1.1 基础接口	89
14.1.2 配置和控制接口	97
14.1.3 监听接口	113
14.1.4 预览接口	115
14.1.5 回放接口	118
14.1.6 语音对讲接口	127
14.1.7 报文传输接口	131
14.2 报警监听模块	140
14.2.1 基础接口	140
14.2.2 配置和控制接口	145
14.2.3 监听接口	147
14.3 取流模块	148
14.3.1 基础接口	148
14.3.2 预览接口	154
14.3.3 回放接口	156
14.3.4 语音对讲接口	159
14.4 存储管理模块	162
14.4.1 基础接口	162
14.4.2 监听接口	168
14.4.3 客户端相关接口	169

第 15 章 回调函数	175
15.1 ASYNC_RESPONSE_CB	175
15.2 DEVICE_REGISTER_CB	175
15.3 ECMSCallback	177
15.4 EHomeMsgCallBack	178
15.5 EHomeSSMsgCallBack	178
15.6 EHomeSSRWCallBack	179
15.7 EHomeSSRWCallBackEx	180
15.8 EHomeSSStorageCallBack	181
15.9 fExceptionCallBack	181
15.10 fVoiceDataCallBack	182
15.11 LongConfigCallBack	183
15.12 PASSTHROUGHDATACALLBACK	184
15.13 PLAYBACK_DATA_CB	185
15.14 PLAYBACK_NEWLINK_CB	185
15.15 PREVIEW_DATA_CB	186
15.16 PREVIEW_NEWLINK_CB	187
15.17 VOICETALK_DATA_CB	187
15.18 VOICETALK_NEWLINK_CB	188
附录 A. 结构体和枚举	342
A.1 Data Structure	342
A.1.1 NET_DVR_TYPE_NAME	342
A.1.2 NET_EHOME_ADB_DEBUG	342
A.1.3 NET_EHOME_ALARM_HOST_LOG	342
A.1.4 NET_EHOME_ALARM_HOST_LOG_COND	343
A.1.5 NET_EHOME_ALARM_INFO	344
A.1.6 NET_EHOME_ALARM_ISAPI_INFO	345
A.1.7 NET_EHOME_ALARM_ISAPI_PICDATA	346

A.1.8 NET_EHOME_ALARM_LISTEN_PARAM	346
A.1.9 NET_EHOME_ALARM_MPDCDATA	347
A.1.10 NET_EHOME_ALARM_MSG	348
A.1.11 NET_EHOME_ALARM_STATUS_UNION	350
A.1.12 NET_EHOME_ALARM_TIME_CFG	350
A.1.13 NET_EHOME_ALARM_TIME_COND	351
A.1.14 NET_EHOME_ALARMIN_CFG	351
A.1.15 NET_EHOME_ALARMIN_COND	352
A.1.16 NET_EHOME_ALARMIN_LINKAGE_TYPE	352
A.1.17 NET_EHOME_ALARMOUT_CFG	353
A.1.18 NET_EHOME_ALARMOUT_STATUS_CFG	354
A.1.19 NET_EHOME_ALARMWIRELESSINFO	354
A.1.20 NET_EHOME_AMS_ADDRESS	354
A.1.21 NET_EHOME_ASYNC_RESP_CB_DATA	355
A.1.22 NET_EHOME_BINARY_RECV_DATA	355
A.1.23 NET_EHOME_BINARY_SEND_DATA	358
A.1.24 NET_EHOME_BLACKLIST_SEVER	361
A.1.25 NET_EHOME_CHAN_STATUS_CHANGED	361
A.1.26 NET_EHOME_CHAN_TIMING_STATUS_SINGLE	362
A.1.27 NET_EHOME_CHAR_EFFECT	362
A.1.28 NET_EHOME_CID_INFO	363
A.1.29 NET_EHOME_CID_INFO_INTERNAL_EX	364
A.1.30 NET_EHOME_CID_INFO_PICTUREINFO_EX	365
A.1.31 NET_EHOME_CID_PARAM	365
A.1.32 NET_EHOME_CMS_LISTEN_PARAM	366
A.1.33 NET_EHOME_CMSCB_DATA	367
A.1.34 NET_EHOME_COMPONENT_GROUP_INFO	370
A.1.35 NET_EHOME_COMPRESSION_CFG	370

A.1.36 NET_EHOME_COMPRESSION_COND	372
A.1.37 NET_EHOME_CONFIG	372
A.1.38 NET_EHOME_DEV_DEFAULT_PARAM	373
A.1.39 NET_EHOME_DEV_LOG	374
A.1.40 NET_EHOME_DEV_LOG_COND	375
A.1.41 NET_EHOME_DEV_REG_INFO	376
A.1.42 NET_EHOME_DEV_REG_INFO_V12	377
A.1.43 NET_EHOME_DEV_SESSIONKEY	378
A.1.44 NET_EHOME_DEV_SESSIONKEY_EHOME50	378
A.1.45 NET_EHOME_DEV_STATUS_CHANGED	378
A.1.46 NET_EHOME_DEV_TIMING_STATUS	379
A.1.47 NET_EHOME_DEV_WORK_STATUS	379
A.1.48 NET_EHOME_DEVICE_CFG	380
A.1.49 NET_EHOME_DEVICE_INFO	381
A.1.50 NET_EHOME_DISCOVERY_MODE	383
A.1.51 NET_EHOME_DISPLAY_PARAM	383
A.1.52 NET_EHOME_ERRCODE_RET	384
A.1.53 NET_EHOME_ETHERNET	385
A.1.54 NET_EHOME_FACESNAP_REPORT	385
A.1.55 NET_EHOME_FINDCOND	387
A.1.56 NET_EHOME_FINDDATA	390
A.1.57 NET_EHOME_FLOW_COND	390
A.1.58 NET_EHOME_FLOW_INFO	391
A.1.59 NET_EHOME_GPS_CFG	391
A.1.60 NET_EHOME_GPS_INFO	392
A.1.61 NET_EHOME_HD_STATUS_CHANGED	393
A.1.62 NET_EHOME_HD_TIMING_STATUS_SINGLE	394
A.1.63 NET_EHOME_HEATMAP_REPORT	394

A.1.64 NET_EHOME_HEATMAP_VALUE	395
A.1.65 NET_EHOME_HTTP_PARAM	396
A.1.66 NET_EHOME_HUMAN_FEATURE	396
A.1.67 NET_EHOME_IDENTIFICATION	397
A.1.68 NET_EHOME_IMAGE_CFG	397
A.1.69 NET_EHOME_INSERT_CHARACTER	398
A.1.70 NET_EHOME_INSERT_EFFECT	398
A.1.71 NET_EHOME_INSERT_INFO	399
A.1.72 NET_EHOME_INSERT_INFO_V20	399
A.1.73 NET_EHOME_INSERT_TEXT_INFO	400
A.1.74 NET_EHOME_IPADDR	401
A.1.75 NET_EHOME_IPADDRESS	401
A.1.76 NET_EHOME_IPC_INFO	401
A.1.77 NET_EHOME_LINKAGE_ALARMOUT	402
A.1.78 NET_EHOME_LINKAGE_PTZ	403
A.1.79 NET_EHOME_LISTEN_PREVIEW_CFG	403
A.1.80 NET_EHOME_LISTEN_VOICETALK_CFG	404
A.1.81 NET_EHOME_LOCAL_ACCESS_SECURITY	404
A.1.82 NET_EHOME_LOCAL_DEV_PINGREO	405
A.1.83 NET_EHOME_LOCAL_GENERAL_CFG	405
A.1.84 NET_EHOME_LOCAL_PLAYBACK_PARAM	406
A.1.85 NET_EHOME_LONG_CFG_INPUT	406
A.1.86 NET_EHOME_LONG_CFG_SEND	406
A.1.87 NET_EHOME_MAKE_I_FRAME	407
A.1.88 NET_EHOME_MANUAL_IOOUT_CTRL	407
A.1.89 NET_EHOME_MPDATA	408
A.1.90 NET_EHOME_MPGPS	408
A.1.91 NET_EHOME_NETWORK_CFG	409

A.1.92 NET_EHOME_NEWLINK_CB_MSG	409
A.1.93 NET_EHOME_NOTICE_PICURL	411
A.1.94 NET_EHOME_NOTIFY_FAIL_INFO	412
A.1.95 NET_EHOME_PASSTHROUGH_PARAM	413
A.1.96 NET_EHOME_PASSWORD_CFG	413
A.1.97 NET_EHOME_PIC_CFG	414
A.1.98 NET_EHOME_PIC_FILE	415
A.1.99 NET_EHOME_PIC_FILE_COND	416
A.1.100 NET_EHOME_PIXEL_ARRAY_SIZE	417
A.1.101 NET_EHOME_PLANCTRLPARAM	418
A.1.102 NET_EHOME_PLAYBACK_DATA_CB_INFO	418
A.1.103 NET_EHOME_PLAYBACK_DATA_CB_PARAM	418
A.1.104 NET_EHOME_PLAYBACK_INFO_IN	419
A.1.105 NET_EHOME_PLAYBACK_INFO_OUT	421
A.1.106 NET_EHOME_PLAYBACK_LISTEN_PARAM	421
A.1.107 NET_EHOME_PLAYBACK_NEWLINK_CB_INFO	422
A.1.108 NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM	423
A.1.109 NET_EHOME_PLAYER_PARAM	424
A.1.110 NET_EHOME_POINT	424
A.1.111 NET_EHOME_POSITION_INFO	425
A.1.112 NET_EHOME_POST_PUBLISH_XML	425
A.1.113 NET_EHOME_POST_SCHEDULE	426
A.1.114 NET_EHOME_PPPOECFG	426
A.1.115 NET_EHOME_PRESET_PARAM	427
A.1.116 NET_EHOME_PREVIEW_CB_MSG	427
A.1.117 NET_EHOME_PREVIEW_DATA_CB_PARAM	428
A.1.118 NET_EHOME_PREVIEWINFO_IN	428
A.1.119 NET_EHOME_PREVIEWINFO_IN_V11	429

A.1.120 NET_EHOME_PREVIEWINFO_OUT	429
A.1.121 NET_EHOME_PT_PARAM	430
A.1.122 NET_EHOME_PTXML_PARAM	430
A.1.123 NET_EHOME_PTZ_PARAM	431
A.1.124 NET_EHOME_PUBLISH_SERVERADDR	433
A.1.125 NET_EHOME_PUSHPLAYBACK_IN	433
A.1.126 NET_EHOME_PUSHPLAYBACK_OUT	434
A.1.127 NET_EHOME_PUSHSTREAM_IN	434
A.1.128 NET_EHOME_PUSHSTREAM_OUT	434
A.1.129 NET_EHOME_PUSHVOICE_IN	435
A.1.130 NET_EHOME_PUSHVOICE_OUT	435
A.1.131 NET_EHOME_PZIN_PARAM	435
A.1.132 NET_EHOME_QUEUE_DATA	436
A.1.133 NET_EHOME_QUEUE_DATA_LIST	436
A.1.134 NET_EHOME_QUEUE_INFO	436
A.1.135 NET_EHOME_QUEUE_ITEM_DATA_LIST	437
A.1.136 NET_EHOME_REC_FILE	437
A.1.137 NET_EHOME_REC_FILE_COND	439
A.1.138 NET_EHOME_RECORD_CHAN	439
A.1.139 NET_EHOME_REGISTER_LISTEN_MODE	440
A.1.140 NET_EHOME_RELATE_IPC_INFO	440
A.1.141 NET_EHOME_RELEASE_SERVER_INFO	441
A.1.142 NET_EHOME_REMOTE_CTRL_PARAM	442
A.1.143 NET_EHOME_REPLACE_MATERIAL	443
A.1.144 NET_EHOME_SCHEDTIME	444
A.1.145 NET_EHOME_SCREEN_SHOT	444
A.1.146 NET_EHOME_SCREEN_SHOT_EX	444
A.1.147 NET_EHOME_SCREEN_SHOT_RET	445

A.1.148 NET_EHOME_SEND_PARAM	445
A.1.149 NET_EHOME_SERVER_INFO	445
A.1.150 NET_EHOME_SERVER_INFO_V50	446
A.1.151 NET_EHOME_SET_REREGISTER_MODE	448
A.1.152 NET_EHOME_SHOT_PIC	448
A.1.153 NET_EHOME_SS_CENTRAL_PARAM	449
A.1.154 NET_EHOME_SS_CLIENT_PARAM	450
A.1.155 NET_EHOME_SS_CLOUD_PARAM	450
A.1.156 NET_EHOME_SS_EX_PARAM	451
A.1.157 NET_EHOME_SS_INIT_PARAM	451
A.1.158 NET_EHOME_SS_KMS_PARAM	452
A.1.159 NET_EHOME_SS_LISTEN_HTTPS_PARAM	452
A.1.160 NET_EHOME_SS_LISTEN_PARAM	453
A.1.161 NET_EHOME_SS_LOCAL_SDK_PATH	454
A.1.162 NET_EHOME_SS_RW_PARAM	455
A.1.163 NET_EHOME_SS_STORAGE_URI	456
A.1.164 NET_EHOME_SS_TOMCAT_MSG	456
A.1.165 NET_EHOME_SS_TOMCAT_PARAM	458
A.1.166 NET_EHOME_SS_VRB_PARAM	459
A.1.167 NET_EHOME_STOPPLAYBACK_PARAM	459
A.1.168 NET_EHOME_STOPSTREAM_PARAM	459
A.1.169 NET_EHOME_STOPVOICETALK_STM_PARAM	460
A.1.170 NET_EHOME_SWITCH_DAILY_PLAN	460
A.1.171 NET_EHOME_SWITCH_PLAN	461
A.1.172 NET_EHOME_SWITCH_PLAN_PARAM	461
A.1.173 NET_EHOME_SWITCH_WEEKLY_PLAN	462
A.1.174 NET_EHOME_SWITCH_DAY_OF_WEEK_PLAN	462
A.1.175 NET_EHOME_SYSCOMPONENT_REDUCED_ADDR	462

A.1.176 NET_EHOME_TERM_DEFAULT_GATEWAY	463
A.1.177 NET_EHOME_TERM_IP_ADDRESS	463
A.1.178 NET_EHOME_TERM_TIME_ZONE	464
A.1.179 NET_EHOME_TERMINAL_CONTROL	464
A.1.180 NET_EHOME_TERMINAL_CONTROL_V20	464
A.1.181 NET_EHOME_TERMINAL_INFO	465
A.1.182 NET_EHOME_TERMINAL_NAME	466
A.1.183 NET_EHOME_TERMINAL_PROGRESS	466
A.1.184 NET_EHOME_TERMINAL_REPORT_INFO	467
A.1.185 NET_EHOME_TERMINAL_STATE	467
A.1.186 NET_EHOME_TERMINAL_UPG_ST_REPORT	468
A.1.187 NET_EHOME_TEXT_MESSAGE	468
A.1.188 NET_EHOME_TIME	469
A.1.189 NET_EHOME_TIME_ADJUST_PARAM	470
A.1.190 NET_EHOME_UPGRADE_CONN_PARAM	470
A.1.191 NET_EHOME_VERSION_INFO	470
A.1.192 NET_EHOME_VOICE_TALK_IN	471
A.1.193 NET_EHOME_VOICE_TALK_OUT	471
A.1.194 NET_EHOME_VOICETALK_DATA	472
A.1.195 NET_EHOME_VOICETALK_DATA_CB_INFO	472
A.1.196 NET_EHOME_VOICETALK_DATA_CB_PARAM	472
A.1.197 NET_EHOME_VOICETALK_NEWLINK_CB_INFO	473
A.1.198 NET_EHOME_VOICETALK_PARA	474
A.1.199 NET_EHOME_VOLUME_DAILY_PLAN	475
A.1.200 NET_EHOME_VOLUME_DAY_OF_WEEK_PLAN	475
A.1.201 NET_EHOME_VOLUME_PLAN	476
A.1.202 NET_EHOME_VOLUME_PLAN_PARAM	476
A.1.203 NET_EHOME_VOLUME_WEEKLY_PLAN	477

A.1.204 NET_EHOME_WEATHER_INFO	477
A.1.205 NET_EHOME_WIRELESS_INFO_CFG	478
A.1.206 NET_EHOME_XML_CFG	478
A.1.207 NET_EHOME_XML_REMOTE_CTRL_PARAM	479
A.1.208 NET_EHOME_ZONE	480
A.2 枚举定义	480
A.2.1 EN_ALARM_TYPE	480
A.2.2 LONG_LINK_MSG	486
A.2.3 NET_CMS_ENUM_PROXY_TYPE	486
A.2.4 NET_EHOME_ADDRESS_TYPE	487
A.2.5 NET_EHOME_ADDRFORMAT_CMD	487
A.2.6 NET_EHOME_CALLBACK_TYPE	488
A.2.7 NET_EHOME_CMS_INIT_CFG_TYPE	489
A.2.8 NET_EHOME_CONTROL_TYPE	489
A.2.9 NET_EHOME_DAY_OF_WEEK1	491
A.2.10 NET_EHOME_EALARM_INIT_CFG_TYPE	491
A.2.11 NET_EHOME_ERROR_CODE	492
A.2.12 NET_EHOME_ESTREAM_INIT_CFG_TYPE	493
A.2.13 NET_EHOME_INSERT_STATE	494
A.2.14 NET_EHOME_IPC_TYPE	494
A.2.15 NET_EHOME_LOCAL_CFG_TYPE	495
A.2.16 NET_EHOME_PLANCTRL_TYPE	496
A.2.17 NET_EHOME_PLAY_SCHEDULE_TYPE1	496
A.2.18 NET_EHOME_PLAYBACK_OPERATE_MODE	497
A.2.19 NET_EHOME_POS_MODE	498
A.2.20 NET_EHOME_POWER_ONOFF	498
A.2.21 NET_EHOME_PROGRAM_INSERT_TYPE	499
A.2.22 NET_EHOME_REFRESH_TYPE	499

A.2.23 NET_EHOME_REGISTER_TYPE	499
A.2.24 NET_EHOME_SS_CLIENT_TYPE	501
A.2.25 NET_EHOME_SS_INIT_CFG_TYPE	502
A.2.26 NET_EHOME_SS_MSG_TYPE	502
A.2.27 NET_EHOME_SS_TYPE	503
A.2.28 NET_EHOME_STREAM_TYPE	504
A.2.29 NET_EHOME_TALK_ENCODING_TYPE	504
A.2.30 NET_EHOME_TERMINAL_PLAY_STATE	505
A.2.31 NET_EHOME_TERMINAL_TYPE	506
A.2.32 NET_EHOME_THIRD_PARTY_DATA_TYPE	507
A.2.33 NET_EHOME_TIME_ZONE	507
A.2.34 NET_EHOME_TRANSMINT_PROTOCOL	508
A.2.35 SEARCH_GET_NEXT_STATUS_ENUM	509
A.2.36 SEARCH_TYPE	510
第 16 章 命令和报文	189
16.1 控制命令报文	189
16.1.1 ADJUSTTIME	189
16.1.2 GETDEVICEWORKSTATUS	189
16.1.3 PZIN	191
16.1.4 REBOOT	192
16.1.5 RESTARTIVSLIB	192
16.1.6 SETDEFENSESTATUS	193
16.1.7 UPDATE	194
16.2 配置命令报文	195
16.2.1 GetAlarmDeploymentTime	195
16.2.2 GetAlarmLinkageType	196
16.2.3 GetAlarmTriggerCapture	197
16.2.4 GetAlarmTriggerRecord	198

16.2.5 GetCalibrationPara	199
16.2.6 GetDevAbility	200
16.2.7 GetExceptionPara	201
16.2.8 GetHideAlarmArea	202
16.2.9 GetHideAlarmPara	203
16.2.10 GetHideArea	203
16.2.11 GetHidePara	205
16.2.12 GetMotionArea	205
16.2.13 GetMotionPara	206
16.2.14 GetPassengerLinePara	207
16.2.15 GetPassengerOSDPara	208
16.2.16 GetPassengerPara	209
16.2.17 GetRecordPlanPara	210
16.2.18 GetRS232Para	211
16.2.19 GetRS485Para	212
16.2.20 GetVILostPara	213
16.2.21 SetAlarmDeploymentTime	214
16.2.22 SetAlarmLinkageType	215
16.2.23 SetAlarmTriggerCapture	216
16.2.24 SetAlarmTriggerRecord	217
16.2.25 SetCalibrationPara	218
16.2.26 SetExceptionPara	219
16.2.27 SetHideAlarmArea	220
16.2.28 SetHideAlarmPara	220
16.2.29 SetHideArea	221
16.2.30 SetHidePara	222
16.2.31 SetMotionArea	223
16.2.32 SetMotionPara	224

16.2.33 SetPassengerLinePara	225
16.2.34 SetPassengerOSDPara	226
16.2.35 SetPassengerPara	227
16.2.36 SetRecordPlanPara	227
16.2.37 SetRS232Para	229
16.2.38 SetRS485Para	230
16.2.39 SetVILostPara	230
16.3 请求 URI	231
16.3.1 /ISAPI/System/capabilities	231
16.3.2 /ISAPI/System/consumptionMode/capabilities?format=json	232
16.3.3 /ISAPI/System/consumptionMode?format=json	232
16.4 XML 或 JSON 报文	233
16.4.1 JSON_Cap_ConsumptionMode	233
16.4.2 JSON_ConsumptionMode	234
16.4.3 JSON_DasInfo	234
16.4.4 JSON_EventNotificationAlert_BatteryStatus	235
16.4.5 JSON_EventNotificationAlert_FaceCaptureMsg	236
16.4.6 JSON_EventNotificationAlert_FaceModelingMsg	244
16.4.7 JSON_EventNotificationAlert_FacePicComparisonMsg	250
16.4.8 JSON_EventNotificationAlert_HeatMapMsg	262
16.4.9 JSON_EventNotificationAlert_mixedTargetDetection	263
16.4.10 JSON_EventNotificationAlert_PeopleQueuingUpAlarmMsg	282
16.4.11 JSON_EventNotificationAlert_UnregisteredStreetVendorMsg	285
16.4.12 JSON_EventNotificationAlert_WaitingTimeDetectionMsg	287
16.4.13 JSON_ResponseStatus	291
16.4.14 XML_DeviceCap	291
16.4.15 XML_EventNotificationAlert_AlarmInputAlarmMsg	306
16.4.16 XML_EventNotificationAlert_ANPRMsg	307

16.4.17 XML_EventNotificationAlert_audioException	313
16.4.18 XML_EventNotificationAlert_DefocusDetectionAlarmMsg	314
16.4.19 XML_EventNotificationAlert_FaceDetectionAlarmMsg	315
16.4.20 XML_EventNotificationAlert_FastMovingDetectionMsg	316
16.4.21 XML_EventNotificationAlert_HDDErrorAlarmMsg	317
16.4.22 XML_EventNotificationAlert_IllegalLoginAlarmMsg	317
16.4.23 XML_EventNotificationAlert_fielddetection	318
16.4.24 XML_EventNotificationAlert_linedetection	320
16.4.25 XML_EventNotificationAlert_LoiteringDetectionAlarmMsg	322
16.4.26 XML_EventNotificationAlert_MotionDetectionMsg	324
16.4.27 XML_EventNotificationAlert_ObjectRemovalDetectionMsg	325
16.4.28 XML_EventNotificationAlert_ParkingDetectionAlarmMsg	326
16.4.29 XML_EventNotificationAlert_PeopleCountingEventMsg	328
16.4.30 XML_EventNotificationAlert_PeopleGatheringDetectionAlarmMsg	329
16.4.31 XML_EventNotificationAlert_PIRAlarmMsg	331
16.4.32 XML_EventNotificationAlert_regionEntrance	332
16.4.33 XML_EventNotificationAlert_regionExiting	334
16.4.34 XML_EventNotificationAlert_sceneChangeDetection	335
16.4.35 XML_EventNotificationAlert_TemperatureAlarmMsg	336
16.4.36 XML_EventNotificationAlert_UnattendedBaggageDetectionAlarmMsg	338
16.4.37 XML_EventNotificationAlert_VideoLossAlarmMsg	339
16.4.38 XML_EventNotificationAlert_VideoTamperingAlarmMsg	340
16.4.39 XML_ResponseStatus	340
附录 B. 附录	511
B.1 事件类型和详情	511
B.2 ISUPSDK 错误码	513
B.3 ISAPI 返回码	523
B.4 功能模块化错误码	570

B.5 设备日志类型	595
B.5.1 报警主机日志类型	612
B.5.2 日志参数配置	622
B.6 支持的网络摄像机	623
B.7 支持的网络硬盘录像机	631
B.8 国家和地区代码	634
B.9 区域码	646

第 1 章 概览

1.1 简介

ISUPSDK 是基于 ISUP（Intelligent Security Uplink Protocol：智能安全上行协议）封装的软件开发工具包，可为第三方平台提供接入海康产品的接口、命令、库文件等。支持的海康产品包括网络摄像机、网络球机、数字视频录像仪（DVR）、网络视频录像仪（NVR）、移动视频录像仪、移动设备、报警设备、门禁设备、解码器等。因此第三方平台可通过该 SDK 实现预览、回放、语音对讲、报警、门禁控制、云台控制等功能。

1.2 系统要求

Windows 和 Linux 操作系统均支持。

对于 32 位 Windows 操作系统

Windows XP/7/8/Vista 和 Windows 服务器 2012/2008/2003

对于 64 位 Windows 操作系统

Windows 7/8/10/和 Windows 服务器 2012/2008

对于 32 位 Linux 操作系统

GCC 版本 4.1.2 及以上

CentOS 版本 5.4 及以上，Redhat AS 版本 5.4 及以上，Ubuntu 版本 9.10 及以上，Fedora 版本 12 及以上

对于 64 位 Linux 操作系统

GCC 版本 4.1.2 及以上

CentOS 版本 5.4 及以上，Redhat AS 版本 5.4 及以上，SUSE 版本 10 及以上

1.3 功能与组件

1.3.1 功能模块

ISUPSDK 包含以下功能模块：预览、回放、文件查找、云台控制、语音对讲、语音转发、报警或事件配置和接收、设备维护、存储管理和参数配置。

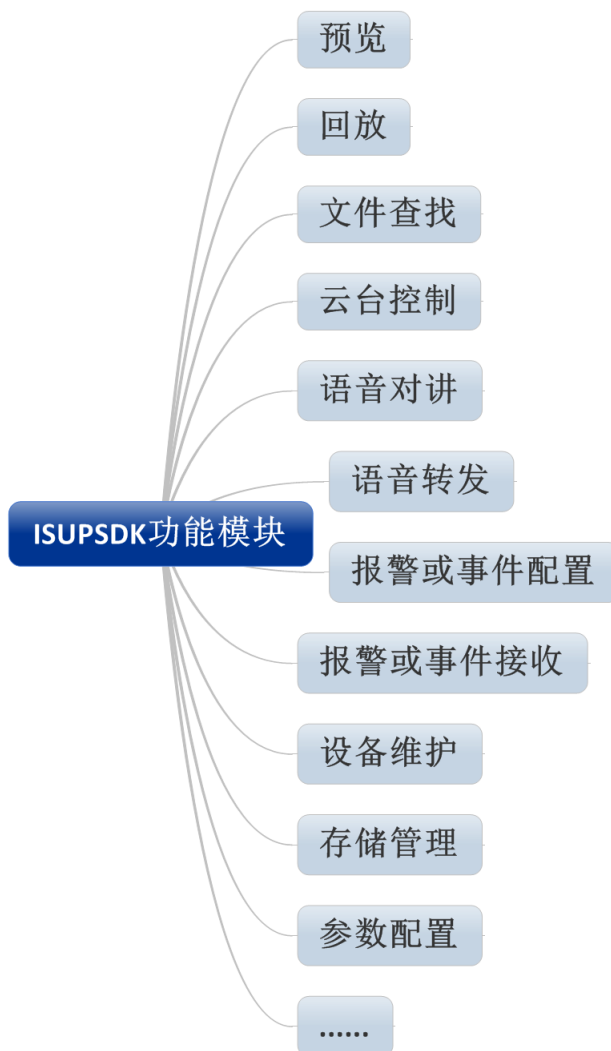


图 1-1 ISUPSDK 功能模块

1.3.2 服务组件

ISUPSDK 由四个服务组件构成，即中心管理服务（Central Management Service，可简称为 CMS）、流媒体服务（Stream Media Service，可简称为 SMS）、报警管理服务（Alarm Management Service，可简称为 AMS）和存储服务（Storage Service，可简称为 SS）。

服务说明

中心管理服务（CMS）

设备主动向 CMS 注册后，可开始预览、文件查找、回放、参数设置、语音对讲等。

流媒体服务（SMS）

可开启预览和回放的监听服务。此外，SMS 的 IP 地址必须通过 CMS 发送给设备。

报警管理服务（AMS）

开启报警监听服务并接收报警信息（如，移动侦测报警、视频丢失报警、视频遮挡报警、行为分析报警、客流统计报警、热度图报警、人脸抓拍报警、GPS 报警等）。此外，AMS 的 IP 地址必须通过 CMS 发送给设备。

存储服务（SS）

可保存、上传和下载图片。

说明

该 SDK 不提供各个服务之间的交互和通信方案，需自行实现。

组件说明

表 1-1 Windows 操作系统

模块名	包含的文件
公共头文件	HCISUPPublic.h
注册模块	HCISUPCMS.dll
	HCISUPCMS.lib
	HCISUPCMS.h
取流模块	HCISUPStream.dll
	HCISUPStream.lib
	HCISUPStream.h
报警监听模块	HCISUPAlarm.dll
	HCISUPAlarm.lib
	HCISUPAlarm.h
存储管理模块	HCISUPSS.dll

模块名	包含的文件
	HCISUPSS.lib
	HCISUPSS.h

表 1-2 Linux 操作系统

模块名	包含的文件
公共头文件	HCISUPPublic.h
注册模块	libHCISUPCMS.so
	HCISUPCMS.h
取流模块	libHCISUPStream.so
	HCISUPStream.h
报警监听模块	libHCISUPAlarm.so
	HCISUPAlarm.h

1.3.3 功能与服务的关系

ISUPSDK 功能模块的实现依赖于服务组件或服务组件之间的合作。

预览和回放

依赖于中心管理服务（CMS）和流媒体服务（SMS）的合作。

文件查找、云台控制、语音转发、设备维护、报警或事件配置和参数配置

依赖于中心管理服务（CMS）。

语音对讲

仅依赖于中心管理服务（CMS）或依赖于中心管理服务（CMS）和流媒体服务（SMS）的合作。

报警或事件接收

依赖于中心管理服务（CMS）和报警管理服务（AMS）的合作。

存储管理

依赖于存储服务（SS）。

1.4 更新历史

版本 V2.5 更新概要_2021/09

修复构造报文引起的异常问题。

版本 V2.5 更新概要_2020/10

1. 扩展初始化配置类型枚举 NET_EHOME_SS_INIT_CFG_TYPE :
新增 NET_EHOME_SS_INIT_CFG_SQLITE3_PATH (设置 sqlite3 库的路径)。
2. 扩展注册回调函数 DEVICE_REGISTER_CB :
新增 6 种回调数据类型类型 : ENUM_DEV_SESSIONKEY_REQ (支持 5.0 版本 ISUP 设备的 SessionKey 请求信息)、ENUM_DEV_DAS_REREGISTER (设备的重注册信息)、ENUM_DEV_DAS_PINGREQ (设备注册心跳)、ENUM_DEV_DAS_EHOMEKEY_ERROR (校验密码失败信息)、ENUM_DEV_SESSIONKEY_ERROR (Sessionkey 交互异常) 和 ENUM_DEV_SLEEP (设备进入休眠状态)。
3. 扩展设备信息结构体 NET_EHOME_DEVICE_INFO :
新增成员 **byStartDTalkChan** (起始数字对讲通道号), 占用 1 个字节。
4. 扩展传输参数结构体 NET_EHOME_PTXML_PARAM :
新增成员 **dwHandle** (消息句柄), 占用 8 个字节。
5. 新增传输 HTTP 页面接口 NET_ECMS_HTTPConfig 。
6. 新增接口用于设置 ISUPSDK 对某个 5.0 协议设备的心跳超时时间 NET_ECMS_SetAliveTimeout 。
7. 扩展 CID 信息结构体 NET_EHOME_CID_INFO :
新增成员 **pPicInfoEx** (附加指针信息) 和 **byRes2** (保留), 占用 8 个字节。
8. 新增附加图片信息结构体 NET_EHOME_CID_INFO_PICTUREINFO_EX 。
9. 扩展 CID 报警扩展信息结构体 NET_EHOME_CID_INFO_INTERNAL_EX :
新增成员 **byVideoType** (视频类型), 占用 16 个字节。
10. 扩展读写扩展回调函数结构体 NET_EHOME_SS_RW_PARAM :
新增成员 **byUseRetIndex** (是否使用上层设置的索引)、**byRes1** (保留)、**pRetIndex** (上层设置的索引), 占用 7 个字节。
11. 扩展设备注册信息结构体 NET_EHOME_DEV_REG_INFO :
扩展成员 **bySupport** (设备支持的功能情况), 占用 1 个字节。
12. 扩展服务器信息结构体 NET_EHOME_SERVER_INFO_V50 :
扩展成员 **dwCloudPoolId** (云存储池 ID), 占用 4 个字节。
13. 扩展文件上传客户端类型枚举 NET_EHOME_SS_CLIENT_TYPE :
新增 NET_EHOME_SS_CLIENT_TYPE_CENTRAL (中心存储流式存储协议)。
14. 扩展回调函数返回的扩展结构体 NET_EHOME_SS_EX_PARAM :

新增联合体成员 `NET_EHOME_SS_CENTRAL_PARAM` (中心存储联合体)。

15. 新增中心存储联合体 `NET_EHOME_SS_CENTRAL_PARAM`。

版本 V2.4 更新概要_2020/04

1. 新增开始预览接口 `NET_ECMS_StartGetRealStream`。
2. 新增预览请求的输入参数结构体 `NET_EHOME_PREVIEWINFO_IN`。
3. 新增开始查找文件接口 `NET_ECMS_StartFindFile`。
4. 新增预览停止请求参数结构体 `NET_EHOME_FINDCOND`。
5. 新增逐个获取检索到的文件接口 `NET_ECMS_FindNextFile`。
6. 新增查询录像文件信息结构体 `NET_EHOME_FINDDATA`。
7. 新增设置初始化参数接口 `NET_ESS_Init_V11`。
8. 新增初始化参数结构体 `NET_EHOME_SS_INIT_PARAM`。
9. 扩展语音对讲请求的输出参数结构体 `NET_EHOME_VOICE_TALK_OUT`：
新增 1 个参数 `IHandle` (输出参数句柄)，占用 4 个字节。
10. 扩展语音对讲请求的输入参数结构体 `NET_EHOME_VOICE_TALK_IN`：
新增 1 个参数 `byEncodingType` (语音对讲编码类型)，占用 9 个字节。
11. 新增语音对讲编码类型枚举 `NET_EHOME_TALK_ENCODING_TYPE`。
12. 扩展设备注册信息结构体 `NET_EHOME_DEV_REG_INFO`：
新增 1 个参数 `byMarketType` (注册类型)，占用 1 个字节。
13. 扩展回调类型枚举 `NET_EHOME_CALLBACK_TYPE`：
新增 2 个类型：`NET_EHOME_PASSTHROUGH_CB` (ISAPI 透传异步回调) 和 `NET_EHOME_V2_VOICETALK_CB` (ISUP2.0 协议语音对讲异步回调)。
14. 扩展中心管理服务服务器监听参数结构体 `NET_EHOME_CMS_LISTEN_PARAM`：
新增 2 个参数：`dwKeepAliveSec` (心跳间隔) 和 `dwTimeOutCount` (心跳超时次数)，共占 8 个字节。
15. 扩展注册回调类型枚举 `NET_EHOME_REGISTER_TYPE`：
新增 2 个类型：`ENUM_DEV_DAS_PINGREO` (设备注册心跳) 和 `ENUM_DEV_SESSIONKEY_ERROR` (Sessionkey 交互异常)。
16. 扩展人体属性信息结构体 `NET_EHOME_HUMAN_FEATURE`：
新增 1 个参数 `byMask` (是否戴口罩)，占用 1 个字节。
17. 扩展关于通过 HTTP 上传的报警图片数据结构体 `NET_EHOME_ALARM_ISAPI_PICDATA`：
新增 1 个参数 `szFilename` (保存图像文件的路径)，占用 8256 字节。
18. 修改设置启用中心管理服务服务器 (CMS) 日志功能接口 `NET_ECMS_SetLogToFile`：
修改参数 `bAutoDel` 描述为：是否在一段时间后自动删除日志文件：“TRUE”-是 (默认)，
“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩 (将这生成的 50 个日志文件压缩成一个压缩文件)。
19. 修改设置启用存储服务器 (SS) 日志功能接口 `NET_ESS_SetLogToFile`：

- 修改参数 **bAutoDel** 描述为：是否在一段时间后自动删除日志文件：“TRUE”-是（默认），“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩（将这生成的 50 个日志文件压缩成一个压缩文件）。
20. 修改设置参数以启用报警管理服务器（AMS）日志功能接口 [NET_EALARM_SetLogToFile](#)：修改参数 **bAutoDel** 描述为：是否在一段时间后自动删除日志文件：“TRUE”-是（默认），“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩（将这生成的 50 个日志文件压缩成一个压缩文件）。
 21. 修改设置启用流媒体服务器（SMS）日志功能接口 [NET_ESTREAM_SetLogToFile](#)：修改参数 **bAutoDel** 描述为：是否在一段时间后自动删除日志文件：“TRUE”-是（默认），“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩（将这生成的 50 个日志文件压缩成一个压缩文件）。
 22. 扩展设置中心管理服务器（CMS）的本地配置参数接口 [NET_ECMS_SetSDKLocalCfg](#) 和获取中心管理服务器（CMS）的本地配置参数接口 [NET_ECMS_GetSDKLocalCfg](#)：新增 2 种本地配置参数类型：DEV_DAS_PINGREO_CALLBACK（是否在注册回调中进行 ENUM_DEV_DAS_PINGREO 回调）和 REGISTER_LISTEN_MODE（CMS 模块的注册监听模式）。
 23. 新增心跳回调设置的结构体 [NET_EHOME_LOCAL_DEV_PINGREO](#)。
 24. 新增注册监听模式结构体 [NET_EHOME_REGISTER_LISTEN_MODE](#)。
 25. 扩展本地配置类型枚举 [NET_EHOME_LOCAL_CFG_TYPE](#)：新增 3 种类型：DEV_DAS_PINGREO_CALLBACK（设备心跳注册回调），REGISTER_LISTEN_MODE（注册监听模式），STREAM_PLAYBACK_PARAM（回放本地参数配置）。
 26. 扩展设置流媒体服务器（SMS）的本地配置参数接口 [NET_ESTREAM_SetSDKLocalCfg](#)：新增 2 种本地配置参数类型：COM_PATH（获取 SMS 模块的 HCApSDKCom（为 HCISUP 相关依赖库的目录）的路径）和 STREAM_PLAYBACK_PARAM（设置是否开启回放同步接收）。
 27. 新增状态类型枚举 [SEARCH_GET_NEXT_STATUS_ENUM](#)。
 28. 扩展终端日志查询结果结构体 [NET_EHOME_DEV_LOG](#)：新增 2 个参数：**byTimeDiffH**（日志时间与国际标准时间（UTC）的小时差）和 **byTimeDiffM**（日志时间与国际标准时间（UTC）的分钟差），共占 2 个字节。
 29. 新增生成存储资源（图片等）的 URL 信息接口 [NET_ESS_GenerateStorageUri](#)。
 30. 新增存储资源的 URI 信息结构体 [NET_EHOME_SS_STORAGE_URI](#)。
 31. 新增存储服务类型枚举 [NET_EHOME_SS_TYPE](#)。
 32. 新增客户端通过内存数据的方式上传资源（图片）接口 [NET_ESS_ClientDoUploadBuffer](#)。
 33. 新增读写扩展回调函数结构体 [NET_EHOME_SS_RW_PARAM](#)。
 34. 新增回调函数返回的扩展结构体 [NET_EHOME_SS_EX_PARAM](#)。
 35. 新增云存储结构体 [NET_EHOME_SS_CLOUD_PARAM](#)。
 36. 新增 TOMCAT 存储信息结构体 [NET_EHOME_SS_TOMCAT_PARAM](#)。
 37. 新增 KMS 存储信息结构体 [NET_EHOME_SS_KMS_PARAM](#)。
 38. 新增 VRB 存储信息结构体 [NET_EHOME_SS_VRB_PARAM](#)。

版本 V2.3.5.1 更新概要_2020/04

相关产品：软件版本为 5.5.110 的网络摄像机（G4 野保），设备型号：DS-2XS6C22F-I(W)(GLG)

1. 扩展被动红外报警报文 [XML_EventNotificationAlert_PIRAlarmMsg](#)：
新增 1 个节点 `<CaptureList>`（抓图列表）。
2. 扩展设备能力报文 [XML_DeviceCap](#)（相关 URI：[/ISAPI/System/capabilities](#)）：
新增 1 个节点 `<isSupportBatteryStatusUpload>`（是否支持电池状态上报）。
新增 2 个节点至节点 `<SysCap>`：`<isSupportConsumptionMode>`（是否支持功耗模式切换）和 `<isSupportManualPowerConsumption>`（是否支持功耗手动控制）。
3. 扩展设备功耗模式能力报文 [JSON_Cap_ConsumptionMode](#)（相关 URI：[/ISAPI/System/consumptionMode/capabilities?format=json](#)）：
新增 1 个功耗模式“sleepOrWakeup”（休眠唤醒模式）至 `devWorkMode`；
新增 1 个节点“sleepOrWakeupPrompt”（设备休眠唤醒模式提示语）。
4. 扩展设备功耗模式参数报文 [JSON_ConsumptionMode](#)（相关 URI：[/ISAPI/System/consumptionMode?format=json](#)）：
新增 1 个功耗模式“sleepOrWakeup”（休眠唤醒模式）至 `devWorkMode`。
5. 新增电池状态事件类型（batteryStatus）及详情 [JSON_EventNotificationAlert_BatteryStatus](#) 至 [事件类型和详情](#)。
6. 扩展注册回调类型枚举 [NET_EHOME_REGISTER_TYPE](#)：
新增 1 个类型：ENUM_DEV_SLEEP（设备进入休眠状态）。
7. 新增将设备从休眠状态唤醒接口 [NET_ECMS_WakeUp](#)，流程详见 [配置功耗模式](#)。
8. 新增 2 个错误码至 [ISUPSDK 错误码](#)：8301-“NET_ERR_DEV_SLEEP”（设备当前处于休眠状态），8302-“NET_ERR_DEV_GOINGTOSLEEP”（即将进入休眠状态，预计 13 秒）。

版本 V2.3.1.2 更新概要_2019/11

相关产品：软件版本为 4.21.005 的 I 系列网络硬盘录像机（具体产品型号详见 [支持的网络硬盘录像机](#)）

添加温度报警（详情参见 [XML_EventNotificationAlert_TemperatureAlarmMsg](#)）、硬盘错误报警（详情参见 [XML_EventNotificationAlert_HDDErrorAlarmMsg](#)）和非法登录报警（详情参见 [XML_EventNotificationAlert_IllegalLoginAlarmMsg](#)）至 [事件类型和详情](#)。

版本 V2.3.1.2 更新概要_2019/11

相关产品：软件版本为 5.6.0 的 50 系列和 70 系列枪机、51 系列和 71 系列室内半球、55 系列和 75 系列室外半球、5A 系列和 7A 系列筒机、7D 系列半球、7T 系列筒机、72 系列枪机、护罩一体机、78 系列筒机、防爆人脸相机（具体产品型号详见 [支持的网络摄像机](#)）

1. 新增命令报文实现云台控制功能，详见 [云台控制](#)。
2. 新增命令报文设置报警或事件参数，详见 [报警或事件配置](#)。
3. 新增多个可接收到的网络摄像机报警或事件类型及详情，详见 [事件类型和详情](#)。
4. 新增设备维护相关命令报文，如升级、获取设备信息、校时、重启等，详见 [设备维护](#)。

5. 新增命令报文配置参数，如网络、编码、图像、OSD、录像、串口等参数，详见 [参数配置](#)。
6. 将预览和回放过程中的异常回调函数注册操作修改为必选操作，详见 [预览](#) 和 [回放](#)。
7. 新增以下 4 个错误码至 [ISUPSDK 错误码](#)：3601-“NET_SS_CLIENT_ERR_KMS_TOKEN_FAIL”、3602-“NET_SS_CLIENT_ERR_KMS_UPLOAD_FAIL”、3603-“NET_SS_CLIENT_ERR_CLOUD_POOLIST_FAIL”、和 3604-“NET_SS_CLIENT_ERR_CLOUD_BESTNODE_FAIL”。
8. 新增存储回调函数（[EHomeSSStorageCallBack](#)）和读写回调函数（[EHomeSSRWCallBack](#)）之间的约束说明，详见结构体 [NET_EHOME_SS_LISTEN_PARAM](#)。

版本 V2.3 更新概要_2019/08

新建文档。

1.5 注意事项

- 在 Windows 32 位或 Linux 32 位操作系统下，中心管理服务（CMS）最多可支持 2048 个设备；在 Windows 64 位或 Linux 64 位操作系统下，中心管理服务（CMS）最多可支持 10,000 个设备。
- 在 Windows 32 位或 Linux 32 位操作系统下，最多可支持 512 路预览或回放；在 Windows 64 位或 Linux 64 位操作系统下，最多可支持 2048 路预览或回放。但对于 ISUP4.0，最多只支持 512 路预览或回放，与操作系统无关。
- 报警管理服务（AMS）每秒最多可接收 2,000 个报警。
- 存储服务（SS）每秒最多可上传或下载 20 张图片，且每张图片大小不超过 200 KB。

第 2 章 设备注册

在通过 ISUPSDK 实现设备功能之前，必须先通过网页登录设备开启 ISUP 功能，并在中心管理服务（CMS）上注册设备。

前提条件

- 确保已调用 `NET_ECMS_Init` 初始化 CMS。
- 确保已调用 `NET_ECMS_StartListen` 开启监听服务。

操作步骤

说明

不同设备的网页配置界面不同，以下步骤仅作参考，具体可参见设备用户手册。

1. 打开浏览器并通过网页登录设备。
 2. 点击 **配置** → **网络** → **高级配置** → **平台接入** 进入平台接入配置界面。
 3. 选择平台接入方式为 **ISUP**。
 4. 勾选 **启用** 开启设备的 ISUP 功能。
 5. 设置 ISUP 版本、服务器地址、端口号、设备 ID 和密钥。
-

说明

密钥只有在 ISUP 版本选择 5.0 时是必须配置的。

6. 注册设备。
-

说明

设备注册消息需要按照一定的逻辑流程进行处理，详见 [设备注册消息处理流程](#)。

2.1 设备注册消息处理流程

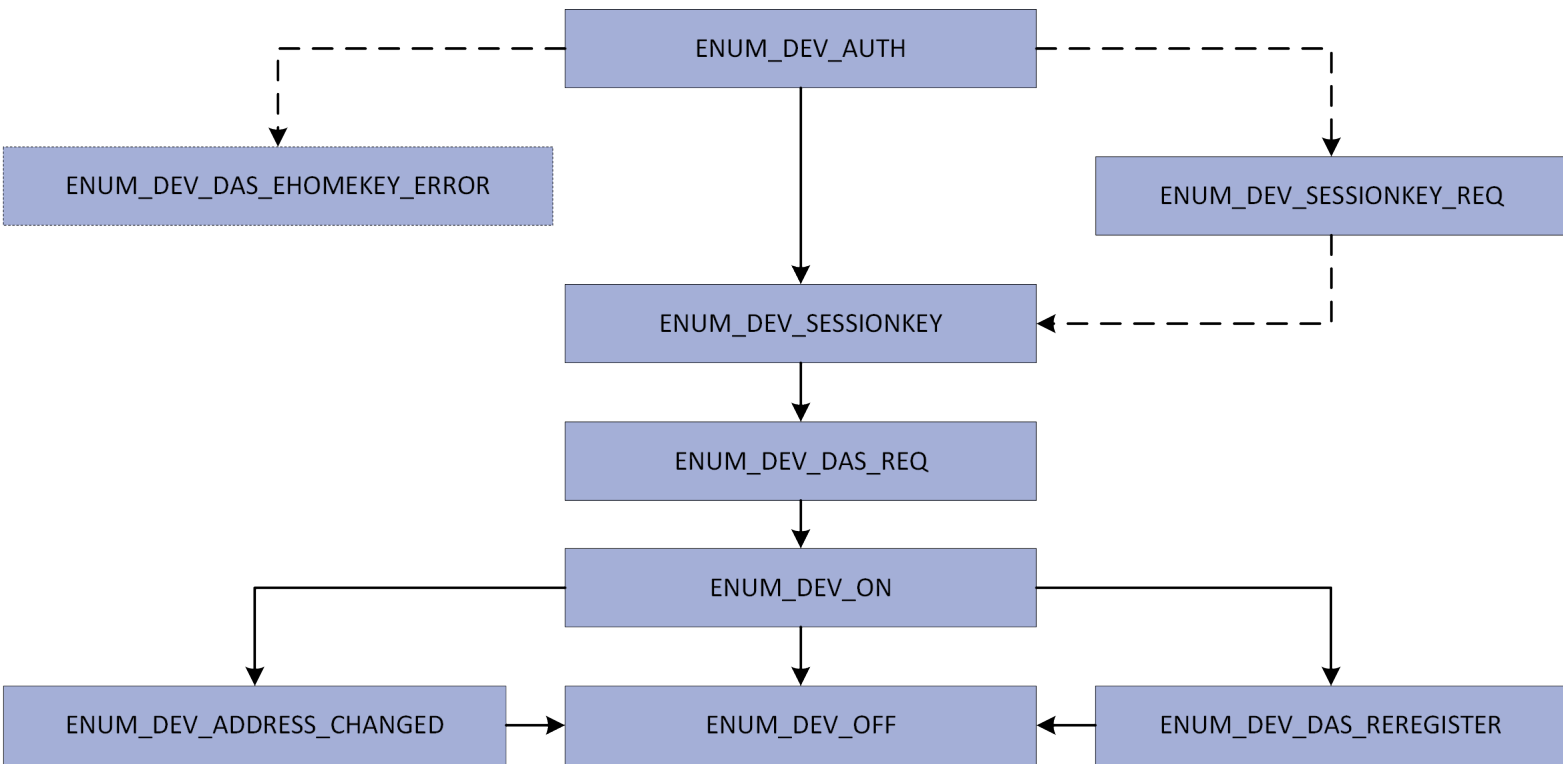


图 2-1 设备注册消息的处理流程图

示例

设备注册消息处理流程的示例代码

```

BOOL _stdcall EHOME_REGISTER(LONG iUserID, DWORD dwDataType, void *pOutBuffer,
DWORD dwOutLen, void *pInBuffer, DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        //设备上线
        //pOutBuffer 指向设备注册信息结构体 NET_EHOME_DEV_REG_INFO_V12
        //pInBuffer 指向服务信息结构体 NET_EHOME_SERVER_INFO
        NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12
*)pOutBuffer;
        NET_EHOME_SERVER_INFO *pServInfo = (NET_EHOME_SERVER_INFO *)pInBuffer;
        //注：结构体内字符串采用 UTF-8 编码方式
        //输出设备信息
        printf("[Device Online] DeviceID[%s]\n", pDevInfo->struRegInfo.byDeviceID);
        //设置报警服务器的 IP 地址
        pServInfo->dwAlarmServerType = 1; //two alarm server types are
supported: TCP and UDP
        //设置 TCP 报警服务器的地址
        strcpy(pServInfo->struTCPAlarmSever.szIP, "10.21.43.15");
        pServInfo->struTCPAlarmSever.wPort = 7332;
    }
}
  
```

```
//设置 UDP 报警服务器的地址
strcpy(pServInfo->struUDPAlarmSever.szIP, "10.21.43.15");
pServInfo->struUDPAlarmSever.wPort = 7332
//注: 根据设备 ID 过滤设备进行注册, 如果设备不允许被注册, 那么会返回 FALSE
return TRUE;
}
else if (ENUM_DEV_OFF == dwDataType)
{
    //设备掉线
    //输出设备信息
    printf("[Device Offline] DeviceID[%s]\n", pDevInfo->struRegInfo.byDeviceID);
    return TRUE;
}
else if (ENUM_DEV_AUTH == dwDataType)
{
    //对于支持 ISUP5.0 设备的认证, 必须配置 EHomeKey
    NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12 *)pOutBuffer;
    //输出设备信息
    printf("[Device Offline] DeviceID[%s] IP[%s]\n", pDevInfo->struRegInfo.byDeviceID, pDevInfo->struRegInfo.struDevAdd.szIP);
    char szEHomeKey[32] = "#Rhg&8D51P";
    strcpy((char*)pInBuffer, szEHomeKey); //Set device EhomeKey for authentication
    return TRUE;
}
else if (ENUM_DEV_SESSIONKEY == dwDataType)
{
    //对于支持 ISUP5.0 设备的认证, 必须配置 EHomeKey
    NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12 *)pOutBuffer;
    printf("[Device SessionKey] DeviceID[%s]\n", pDevInfo->struRegInfo.byDeviceID);
    NET_EHOME_DEV_SESSIONKEY struSessionkey = {0};
    memcpy(struSessionkey.sDeviceID, pDevInfo->struRegInfo.byDeviceID, MAX_DEVICE_ID_LEN);
    memcpy(struSessionkey.sSessionKey, pDevInfo->struRegInfo.bySessionKey, MAX_MASTER_KEY_LEN);
    NET_ECMS_SetDeviceSessionKey(&struSessionkey);

    //对于支持 ISUP5.0 的设备, 报警模块也需要先认证, 因此必须要有设备 SessionKey
    //如果 AMS 和 CMS 安装在同一个程序中, 请直接调用 NET_EALARM_SetDeviceSessionKey
    NET_EALARM_SetDeviceSessionKey(&struSessionkey);
    //如果 AMS 和 CMS 未安装在同一个程序中, 请先将 SessionKey 传输给 AMS, 然后再添加
    return TRUE;
}
else if (ENUM_DEV_DAS_REQ == dwDataType)
{
    //在注册期间, 设备请求获取 DAS 信息
    char szDASIP[128] = "10.21.43.15"; //DAS IP address
```

```
    DWORD dwPort = 6666; //DAS port No.
    sprintf((char*)pInBuffer, "{\\"Type\\":\\"DAS\\",\\"DasInfo\\":{\\"Address\\":
\\"%s\\",
    \\"Domain\\":\\"test.ys7.com\\",\\"ServerID\\":\\"das_%s_%d\\",\\"Port\\":
%d,\\"UdpPort\\":%d}"
        , szLocalIP, szLocalIP, dwPort, dwPort, dwPort);
    }
    else if (ENUM_DEV_ADDRESS_CHANGED == dwDataType)
{
    //设备 IP 地址变化 (比如, 当设备在移动网络中或跨基站时)
    NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12
*)pOutBuffer;
    //输出设备信息
    printf("[Device IP Changed] DeviceID[%s] New IP[%s]\n", pDevInfo-
>struRegInfo.byDeviceID, pDevInfo->struRegInfo.struDevAdd.szIP);
    }
    else if (ENUM_DEV_DAS_REREGISTER == dwDataType)
    {
        //当设备信息发生变化时, 重新注册已注册的设备
        NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12
*)pOutBuffer;
        //输出设备信息
        printf("[Device ReRegister] DeviceID[%s] IP[%s]\n", pDevInfo-
>struRegInfo.byDeviceID, pDevInfo->struRegInfo.struDevAdd.szIP);
        return TRUE;
    }
    return TRUE;
}
```

第 3 章 预览

通过预览可远程获取监控区域的实时视频或音频信息，从而获知人、车、物体等对象的状态。通过 ISUPSDK 集成预览功能需要依赖中心管理服务（CMS）、流媒体服务（SMS）和播放库。

前提条件

- 请确保已调用 *NET_ECMS_Init* 和 *NET_ESTREAM_Init* 初始化 CMS 和 SMS。
- 请确保已调用 *NET_ECMS_StartListen* 开启 CMS 监听服务以便接收设备注册信息。

操作步骤

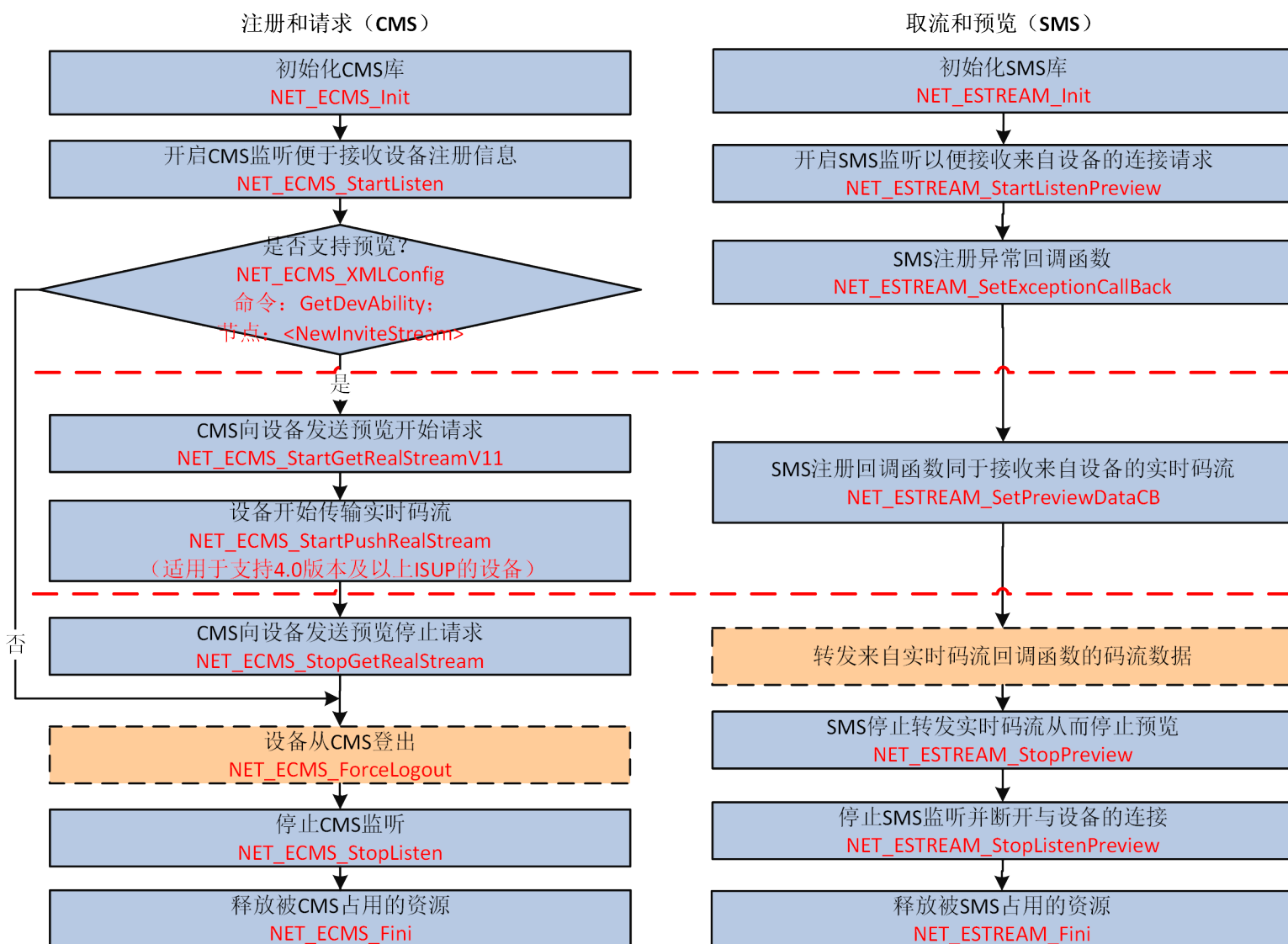


图 3-1 预览的接口调用流程图

1. 调用 `NET_ECMS_XMLConfig` 和命令 `GetDevAbility` 获取设备能力并判断是否支持预览。
设备能力集由 `pOutBuf` 返回。
如果支持, 将返回节点 `<NewInviteStream>`, 可继续下面步骤。
否则, 请结束该任务。
2. 调用 `NET_ESTREAM_StartListenPreview` 开启 SMS 监听服务并接收来自设备的连接请求。
3. 调用 `NET_ESTREAM_SetExceptionCallBack` 为 SMS 注册异常回调函数。
异常信息会被回调至该注册的回调函数中, 请及时处理异常避免阻塞。

4. 调用 `NET_ECMS_StartGetRealStreamV11` 将预览开始请求从 CMS 发送给设备。
发送 SMS 的地址和端口号给设备，同时会自动为 CMS 分配一个会话 ID。
5. 调用 `NET_ESTREAM_SetPreviewDataCB` 为 SMS 注册回调函数用于接收来自设备的实时流。
6. 调用 `NET_ECMS_StartPushRealStream` 将实时流传输请求从 CMS 发送给设备。
设备自动连接 SMS 并开始发送实时流给 SMS。
7. 可选操作: 转发来自 SMS 实时流回调函数的码流。

说明

预览码流的解码和显示由客户端软件实现。

8. 调用 `NET_ESTREAM_StopPreview` 停止 SMS 的实时流转发。
9. 调用 `NET_ECMS_StopGetRealStream` 将预览停止请求从 CMS 发送给设备。
设备停止传输实时码流。
10. 调用 `NET_ESTREAM_StopListenPreview` 停止 SMS 监听服务并断开其与设备的连接。
11. 调用 `NET_ECMS_ForceLogout` 强制注销设备。

示例

预览的示例代码

说明

在该示例代码中，CMS 和 SMS 被安装在同一台电脑的同一个程序中。

```
#include <stdio.h>
#include <Windows.h>
#include "HCISUPCMS.h"
#include "HCISUPStream.h"
#include "plaympeg4.h"

LONG lLoginID = -1;
LONG lLinkHandle = -1;
LONG lRealHandle = -1;
FILE *Videofile = NULL;

////////////////////////////////////
//注册回调函数
BOOL CALLBACK RegisterCallBack(LONG lUserID, DWORD dwDataType, void
*pOutBuffer, DWORD dwOutLen, void *pInBuffer, DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDevInfo = (NET_EHOME_DEV_REG_INFO *)pOutBuffer;

        if (pDevInfo != NULL)
        {
            lLoginID = lUserID;
            printf("On-line, lUserID: %d, Device ID: %s\n", lLoginID, pDevInfo-
```

```
>byDeviceID);
    }
    //输入参数
    NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO *)pInBuffer;
    pServerInfo->dwTimeOutCount = 6; //心跳超时次数
    pServerInfo->dwKeepAliveSec = 15; //心跳间隔
}
else if (ENUM_DEV_OFF == dwDataType)
{
    printf("Off-line, lUserID: %d\n", lUserID);
    NET_ECMS_ForceLogout(lUserID);
}
else if (ENUM_DEV_AUTH == dwDataType)
{
    //对于支持 ISUP5.0 的设备认证, 必须设置认证密钥
    NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12
*)pOutBuffer;
    //输出设备信息
    printf("[Device Auth] DeviceID[%s] IP[%s]\n", pDevInfo->
>struRegInfo.byDeviceID, pDevInfo->struRegInfo.struDevAdd.szIP);
    char szEHomeKey[32] = "i#Rhg&8D51P"; //假设密钥为 i#Rhg&8D51P
    strcpy((char*)pInBuffer, szEHomeKey); //Set device EhomeKey for
authentication
}
else if (ENUM_DEV_SESSIONKEY == dwDataType)
{
    //对于支持 ISUP5.0 的设备认证, 必须配置 EHomeKey
    NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12
*)pOutBuffer;
    printf("[Device SessionKey] DeviceID[%s]\n", pDevInfo->
>struRegInfo.byDeviceID);
    NET_EHOME_DEV_SESSIONKEY struSessionkey = {0};
    memcpy(struSessionkey.sDeviceID, pDevInfo->struRegInfo.byDeviceID,
MAX_DEVICE_ID_LEN);
    memcpy(struSessionkey.sSessionKey, pDevInfo->struRegInfo.bySessionKey,
MAX_MASTER_KEY_LEN);
    NET_ECMS_SetDeviceSessionKey(&struSessionkey);
}
else
{
    //...
}
return TRUE;
}

////////////////////////////////////
//处理实时码流数据
BOOL InputStreamData(BYTE byDataType, char* pBuffer, int iDataLen)
{
    if(Videofile == NULL)
    {
```

```
Videofile = fopen("Test.mp4","wb");
printf("Save data to file: Test.mp4!\n");
}

if(Videofile!= NULL)
{
    fwrite(pBuffer,iDataLen,1,Videofile); //回调实时码流并保存视频文件
}

//调用播放库解码并显示码流实现预览
/*
if(1 == byDataType)
{
    if (!PlayM4_GetPort(&m_lPort))
    {
        return FALSE;
    }
    if (!PlayM4_SetStreamOpenMode(m_lPort, STREAME_REALTIME))
    {
        return FALSE;
    }
    //输入头部的前 40 字节
    if(!PlayM4_OpenStream(m_lPort, (unsigned char *)pBuffer,
(DWORD)iDataLen, 2*1024*1024))
    {
        return FALSE;
    }
    if(!PlayM4_Play(m_lPort, hWnd))
    {
        return FALSE;
    }
}
else
{
    for (int i=0; i<1000; i++)
    {
        BOOL bRet = PlayM4_InputData(m_lPort, (unsigned char *)pBuffer,
(DWORD)iDataLen);
        if (!bRet)
        {
            if ( i >=999)
            {
                printf("PlayM4_InputData failed, error code: %d!",
PlayM4_GetLastError(m_lPort));
            }
            Sleep(2);
        }
        else
        {
            break;
        }
    }
}
```

```
}
*/
return TRUE;
}

////////////////////////////////////
//注册实时码流回调函数
void CALLBACK fnPREVIEW_DATA_CB(LONG lPreviewHandle, NET_EHOME_PREVIEW_CB_MSG
*pPreviewCBMsg, void *pUserData)
{
    if (NULL == pPreviewCBMsg)
    {
        return ;
    }
    lRealHandle = lPreviewHandle;
    InputStreamData (pPreviewCBMsg->byDataType, (char*)pPreviewCBMsg->pRecvdata,
pPreviewCBMsg->dwDataLen);
}

////////////////////////////////////
//注册预览请求的响应回调函数
BOOL CALLBACK fnPREVIEW_NEWLINK_CB(LONG lPreviewHandle,
NET_EHOME_NEWLINK_CB_MSG *pNewLinkCBMsg, void *pUserData)
{
    lLinkHandle = lPreviewHandle;
    printf("Callback of preview listening, Device ID: %s, Channel: %d\n",
pNewLinkCBMsg->szDeviceID, pNewLinkCBMsg->dwChannelNo);

    //预览数据的回调参数
    NET_EHOME_PREVIEW_DATA_CB_PARAM struDataCB = {0};
    struDataCB.fnPreviewDataCB = fnPREVIEW_DATA_CB;
    struDataCB.byStreamFormat = 0;//封装格式:0-PS 格式

    if (!NET_ESTREAM_SetPreviewDataCB(lPreviewHandle, &struDataCB))
    {
        printf("NET_ESTREAM_SetPreviewDataCB failed, error code: %d\n",
NET_ESTREAM_GetLastError());
        return FALSE;
    }
    printf("NET_ESTREAM_SetPreviewDataCB!\n");

    return TRUE;
}

////////////////////////////////////
//设置异常回调
NET_ESTREAM_SetExceptionCallBack(0, 0, StreamExceptionCallback, NULL);

void CALLBACK StreamExceptionCallback(DWORD dwType, LONG iUserID, LONG iHandle,
void* pUser)
{
    if (EHOME_PREVIEW_EXCEPTION == dwType)
```

```
{
    //预览异常
    printf("Preview exception, handle=%d, Error:%d", iHandle, dwError);
}
else if (EHOME_PLAYBACK_EXCEPTION == dwType)
{
    //回放异常
    printf("Playback exception, handle=%d, Error:%d", iHandle, dwError);
}
else if (EHOME_AUDIOTALK_EXCEPTION == dwType)
{
    //语音对讲 (转发) 异常
    printf("Stream VoiceTalk exception, handle=%d, Error:%d", iHandle,
dwError);
}
}

void main(){
    //SMS 在监听服务开启后获取码流

    //初始化 SMS 库
    NET_ESTREAM_Init();

    //预览的监听参数
    NET_EHOME_LISTEN_PREVIEW_CFG struListen = {0};
    memcpy(struListen.struIPAdress.szIP, "10.16.2.123", sizeof("10.16.2.123"));
    struListen.struIPAdress.wPort = 8003; //SMS 的监听端口号
    struListen.fnNewLinkCB = fnPREVIEW_NEWLINK_CB; //预览请求回调函数
    struListen.pUser = NULL;
    struListen.byLinkMode = 1; //0-TCP, 1-UDP

    //开启监听服务
    LONG lHandle = NET_ESTREAM_StartListenPreview(&struListen);
    if(lHandle < -1)
    {
        printf("NET_ESTREAM_StartListenPreview failed, error code: %d\n",
NET_ESTREAM_GetLastError());
        NET_ESTREAM_Fini();
        return;
    }
    printf("NET_ESTREAM_StartListenPreview!\n");
    //注册和预览请求

    //初始化 CMS 库
    NET_ECMS_Init();

    //注册的监听参数
```

```
NET_EHOME_CMS_LISTEN_PARAM struCMSListenPara = {0};
memcpy(struCMSListenPara.struAddress.szIP, "0.0.0.0", sizeof("0.0.0.0"));
struCMSListenPara.struAddress.wPort = 7660;
struCMSListenPara.fnCB = RegisterCallBack;

//开启监听并接收设备注册信息
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
if(lListen < -1)
{
    printf("NET_ECMS_StartListen failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartListen!\n");

while(1)
{
    Sleep(1000); //The following operations should be done when the
registration is completed
    if(lLoginID >= 0)
    {
        break;
    }
}

//预览请求的输入参数
NET_EHOME_PREVIEWINFO_IN_V11 struPreviewIn = {0};
struPreviewIn.iChannel = 1; //通道号
struPreviewIn.dwLinkMode = 1; //0-TCP, 1-UDP
struPreviewIn.dwStreamType = 0; //码流类型：0-主码流, 1-子码流 2-第三码流
memcpy(struPreviewIn.struStreamSever.szIP, "10.16.2.123",
sizeof("10.16.2.123")); //SMS的IP地址
struPreviewIn.struStreamSever.wPort = 8003; //SMS的端口号, 需和监听端口号一致

//预览请求的输出参数
NET_EHOME_PREVIEWINFO_OUT struPreviewOut = {0};

//预览请求
if(!NET_ECMS_StartGetRealStreamV11(lLoginID, &struPreviewIn,
&struPreviewOut))
{
    printf("NET_ECMS_StartGetRealStreamV11 failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartGetRealStreamV11!\n");

//码流传输请求的输入参数
NET_EHOME_PUSHSTREAM_IN struPushStreamIn = {0};
```

```
struPushStreamIn.dwSize = sizeof(struPushStreamIn);
struPushStreamIn.lSessionID = struPreviewOut.lSessionID; //预览请求的会话 ID

//码流传输请求的输出参数
NET_EHOME_PUSHSTREAM_OUT struPushStreamOut = {0};

//发送请求给设备并开始传输实时码流
if(!NET_ECMS_StartPushRealStream(lLoginID, &struPushStreamIn,
&struPushStreamOut))
{
    printf("NET_ECMS_StartPushRealStream failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartPushRealStream!\n");

Sleep(50000);

////////////////////////////////////
//登出

//释放被 CMS 预览请求占用的资源
if(!NET_ECMS_StopGetRealStream(lLoginID, struPreviewOut.lSessionID))
{
    printf("NET_ECMS_StopGetRealStream failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//CMS 停止监听服务
if(!NET_ECMS_StopListen(lListen))
{
    printf("NET_ECMS_StopListen failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//释放被 CMS 占用的资源
NET_ECMS_Fini();
////////////////////////////////////

//SMS 停止转发码流
if(lRealHandle >= 0)
{
    if (!NET_ESTREAM_StopPreview(lRealHandle))
    {
        printf("NET_ESTREAM_StopPreview failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}

//SMS 停止监听服务
```



```
if(lHandle >= 0)
{
    if (!NET_ESTREAM_StopListenPreview(lHandle))
    {
        printf("NET_ESTREAM_StopListenPreview failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}
//释放被 SMS 占用的资源
NET_ESTREAM_Fini();

//释放文件资源
if(Videofile != NULL)
{
    fclose(Videofile);
    Videofile = NULL;
}

printf("Exit!\n");
}
```

后续处理

- 调用 NET_ECMS_StopListen 停止 CMS 监听服务。
- 调用 NET_ECMS_Fini 和 NET_ESTREAM_Fini 释放被 CMS 和 SMS 占用的资源。

第 4 章 文件查找

可通过中心管理服务（CMS）搜索保存在设备存储中的视频或音频文件用于回放。

前提条件

- 请确保已调用 `NET_ECMS_Init` 初始化 CMS。
- 请确保已调用 `NET_ECMS_StartListen` 开启 CMS 监听服务以接收设备注册信息。

操作步骤

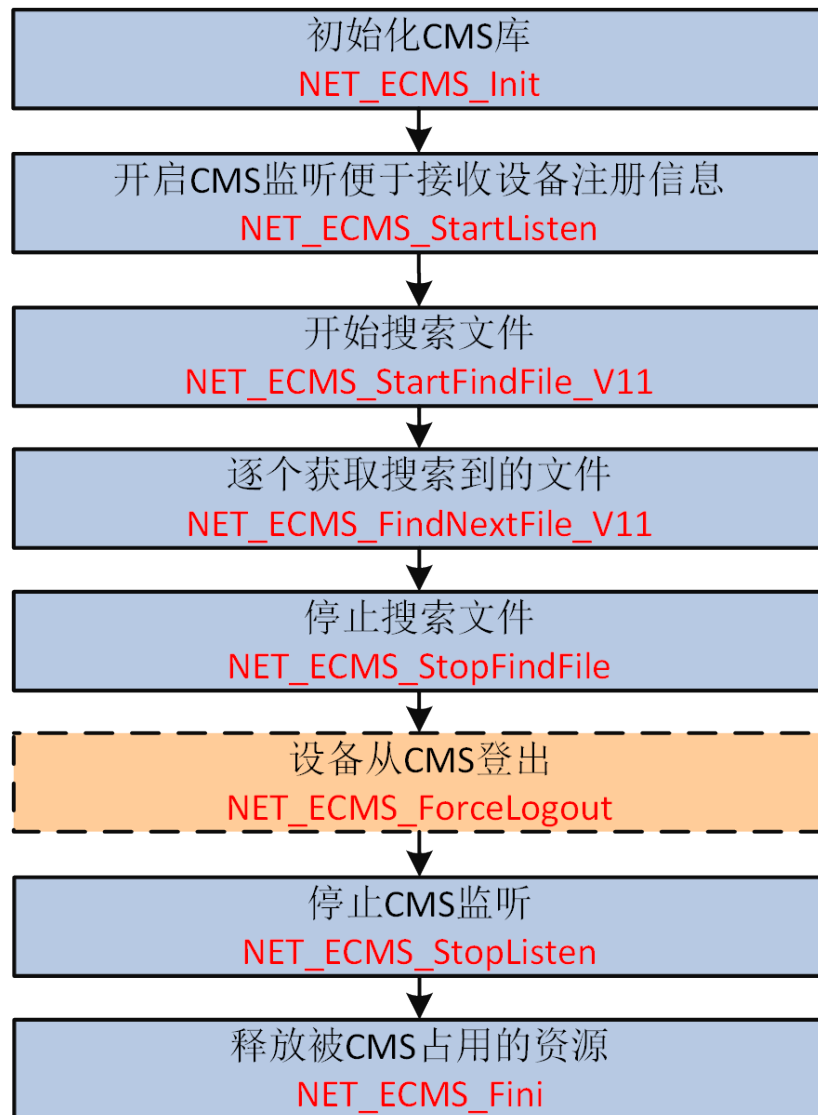


图 4-1 文件查找的接口调用流程图

1. 调用 `NET_ECMS_StartFindFile_V11` 开始查找文件。
2. 调用 `NET_ECMS_FindNextFile_V11` 逐个获取搜索到的文件。
3. 调用 `NET_ECMS_StopFindFile` 停止查找文件。
4. 可选操作: 调用 `NET_ECMS_ForceLogout` 强制注销设备。

示例

文件查找的示例代码

```
#include <stdio.h>
#include <Windows.h>
#include "HCISUPCMS.h"
#include "HCISUPStream.h"
#include "plaympeg4.h"

LONG lLoginID = -1;
LONG lLinkHandle = -1;
LONG lPlayHandle = -1;
FILE *Videofile = NULL;

//初始化 CMS
NET_ECMS_Init();

//注册的监听参数
NET_EHOME_CMS_LISTEN_PARAM struCMSListenPara = {0};
memcpy(struCMSListenPara.struAddress.szIP, "0.0.0.0", sizeof("0.0.0.0"));
struCMSListenPara.struAddress.wPort = 7660;
struCMSListenPara.fnCB = RegisterCallBack;

//开启监听服务并接收设备注册信息
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
if(lListen < -1)
{
    printf("NET_ECMS_StartListen failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartListen!\n");

while(1)
{
    Sleep(1000); //以下操作需在注册完整后才能进行
    if(lLoginID >= 0)
    {
        break;
    }
}

//查找视频文件
NET_EHOME_REC_FILE_COND struFindCond = {0};
```

```
struFindCond.dwChannel = 1; //通道号, 从 1 开始
struFindCond.dwRecType = 0xff; //所有类型
struFindCond.dwStartIndex = 0; //搜索起始位置
struFindCond.dwMaxFileCountPer = 5; //单次可搜索的最多文件数量

//搜索开始时间
struFindCond.struStartTime.wYear = (WORD)2015;
struFindCond.struStartTime.byMonth = (BYTE)5;
struFindCond.struStartTime.byDay = (BYTE)18;
struFindCond.struStartTime.byHour = (BYTE)10;
struFindCond.struStartTime.byMinute = (BYTE)0;
struFindCond.struStartTime.bySecond = (BYTE)0;

//搜索结束时间
struFindCond.struStopTime.wYear = (WORD)2015;
struFindCond.struStopTime.byMonth = (BYTE)5;
struFindCond.struStopTime.byDay = (BYTE)18;
struFindCond.struStopTime.byHour = (BYTE)12;
struFindCond.struStopTime.byMinute = (BYTE)59;
struFindCond.struStopTime.bySecond = (BYTE)59;

LONG lSearchType = 0; //查找视频文件
LONG lFileHandle = NET_ECMS_StartFindFile_V11(lLoginID, lSearchType,
&struFindCond, sizeof(struFindCond)); //开始文件查找
if (lFileHandle < 0)
{
    printf("NET_ECMS_StartFindFile_V11 failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartFindFile_V11!\n");

LONG lRet = -1;
char csTmp[256] = {0};
char szFileName[MAX_FILE_NAME_LEN] = {0};

NET_EHOME_REC_FILE struFileInfo = {0};

//可在线程中查找文件
while(1)
{
    lRet = NET_ECMS_FindNextFile_V11(lFileHandle, &struFileInfo,
sizeof(struFileInfo)); //逐个获取搜索结果
    if (lRet == ENUM_GET_NEXT_STATUS_SUCCESS)
    {
        if (struFileInfo.dwFileSize / 1024 == 0)
        {
            sprintf(csTmp, "%d", struFileInfo.dwFileSize);
        }
        else if (struFileInfo.dwFileSize / 1024 > 0 && struFileInfo.dwFileSize /
```

```
(1024*1024) == 0)
{
    sprintf(csTmp, "%dK", struFileInfo.dwFileSize/1024);
}
else
{
    sprintf(csTmp, "%dM", struFileInfo.dwFileSize /1024/1024);
}

printf("Filename[%s], Filesize[%s], StarTime[%04d-%02d-%02d %02d:%02d:
%02d], StopTime[%04d-%02d-%02d %02d:%02d:%02d] \n", \
    struFileInfo.szFileName, csTmp, struFileInfo.struStartTime.wYear,
struFileInfo.struStartTime.byMonth, \
    struFileInfo.struStartTime.byDay, struFileInfo.struStartTime.byHour,
struFileInfo.struStartTime.byMinute, \
    struFileInfo.struStartTime.bySecond,
struFileInfo.struStopTime.byDay, struFileInfo.struStopTime.byHour, \
    struFileInfo.struStopTime.byMinute,
struFileInfo.struStopTime.bySecond); //搜索到的文件信息

    memcpy(szFileName, struFileInfo.sFileName, MAX_FILE_NAME_LEN);
}
else
{
    if (lRet == ENUM_GET_NETX_STATUS_NEED_WAIT)
    {
        Sleep(5);
        continue;
    }

    if ((lRet == ENUM_GET_NETX_STATUS_NO_FILE) || (lRet ==
ENUM_GET_NEXT_STATUS_FINISH))
    {
        printf("No more file!\n");
        break;
    }
    else if(lRet == ENUM_GET_NEXT_STATUS_NOT_SUPPORT)
    {
        printf("Device does not support!\n");
        break;
    }
    else
    {
        printf("Failed to find a file, for the server is busy or network
failure!\n");
        break;
    }
}
}
//CMS 停止监听服务
if (!NET_ECMS_StopListen (lListen))
{
```

```
    printf("NET_ECMS_StopListen failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//释放被 CMS 占用的资源
NET_ECMS_Fini();
```

后续处理

- 调用 `NET_ECMS_StopListen` 停止 CMS 监听服务。
- 调用 `NET_ECMS_Fini` 释放被 CMS 占用的资源。

第 5 章 回放

回放即播放已存储并搜索到的视频文件。通过 ISUPSDK 实现回放需依赖于中心管理服务（CMS）、流媒体服务（SMS）和播放库。

前提条件

- 请确保已调用 *NET_ECMS_Init* 和 *NET_ESTREAM_Init* 初始化 CMS 和 SMS。
- 请确保已调用 *NET_ECMS_StartListen* 开启 CMS 监听服务以便接收设备注册信息。

操作步骤

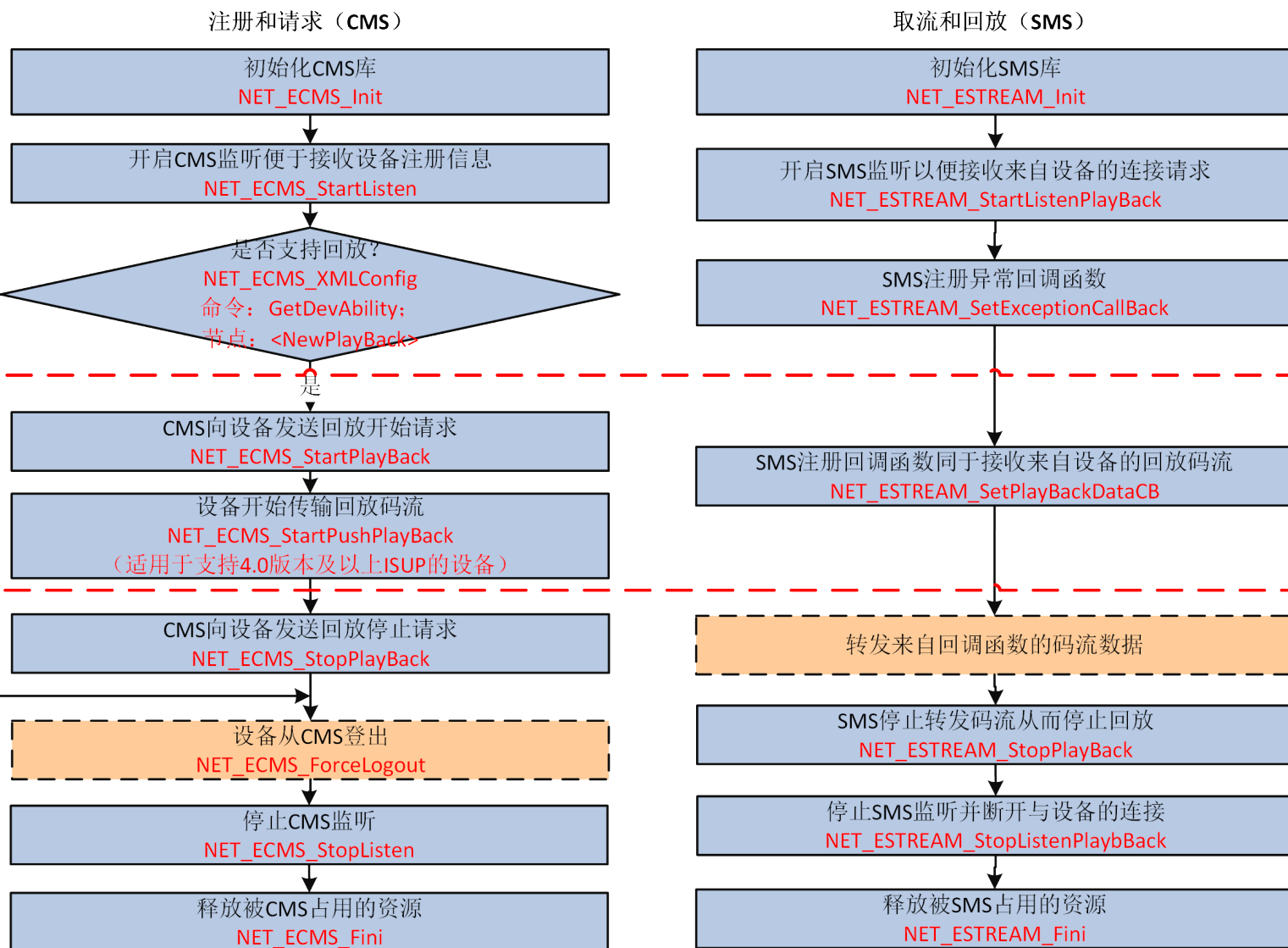


图 5-1 回放的接口调用流程图

1. 调用 `NET_ECMS_XMLConfig` 和命令 `GetDevAbility` 获取设备能力并判断是否支持回放。
设备能力集由 `pOutBuf` 返回。
如果支持, 将返回节点`<NewPlayBack>`, 可继续下面步骤。
否则, 请结束该任务。
2. 调用 `NET_ESTREAM_StartListenPlayBack` 开启 SMS 监听服务并接收来自设备的连接请求。
3. 调用 `NET_ESTREAM_SetExceptionCallBack` 为 SMS 注册异常回调函数。
异常信息会被回调至该注册的回调函数中, 请及时处理异常避免阻塞。

4. 调用 [NET_ECMS_StartPlayBack](#) 将回放开始请求从 CMS 发送给设备。

发送 SMS 的地址和端口号给设备，并会自动为 CMS 分配一个会话 ID。

5. 调用 [NET_ESTREAM_SetPlayBackDataCB](#) 为 SMS 注册回调函数用于接收来自设备的回放码流。

6. 调用 [NET_ECMS_StartPushPlayBack](#) 将码流传输请求从 CMS 发送给设备。

设备将自动连接 SMS 并开始发送码流给 SMS。

7. 可选操作: 转发来自 SMS 码流回调函数的码流。

说明

回放码流的解码和显示由客户端软件实现。

8. 调用 [NET_ESTREAM_StopPlayBack](#) 停止 SMS 的码流转发。

9. 调用 [NET_ECMS_StopPlayBack](#) 将回放停止请求从 CMS 发送给设备。

设备停止传输实时码流。

10. 调用 [NET_ESTREAM_StopListenPlayBack](#) 停止 SMS 监听服务并断开其与设备的连接。

11. 可选操作: 调用 [NET_ECMS_ForceLogout](#) 强制注销设备。

示例

回放的示例代码

说明

在该示例代码中，CMS 和 SMS 被安装在同一台电脑的同一个程序中。

```
#include <stdio.h>
#include <Windows.h>
#include "HCISUPCMS.h"
#include "HCISUPStream.h"
#include "plaympeg4.h"

LONG lLoginID = -1;
LONG lLinkHandle = -1;
LONG lPlayHandle = -1;
FILE *Videofile = NULL;

////////////////////////////////////
//注册回调函数
BOOL CALLBACK RegisterCallBack(LONG lUserID, DWORD dwDataType, void
*pOutBuffer, DWORD dwOutLen, void *pInBuffer, DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDevInfo = (NET_EHOME_DEV_REG_INFO *)pOutBuffer;

        if (pDevInfo != NULL)
        {
            lLoginID = lUserID;
        }
    }
}
```

```
    printf("On-line, lUserID: %d, Device ID: %s\n", lLoginID, pDevInfo->byDeviceID);
    }
    //输入参数
    NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO *)pInBuffer;
    pServerInfo->dwTimeOutCount = 6; //心跳超时次数
    pServerInfo->dwKeepAliveSec = 15; //心跳间隔
    }
    else if (ENUM_DEV_OFF == dwDataType)
    {
    printf("Off-line, lUserID: %d\n", lUserID);
    NET_ECMS_ForceLogout(lUserID);
    }
    else
    {
    }

    return TRUE;
}

////////////////////////////////////
//处理文件流数据
BOOL InputStreamData(BYTE byDataType, char* pBuffer, int iDataLen)
{
    if(Videofile == NULL)
    {
        Videofile = fopen("Test.mp4","wb");
        printf("Save data to file: Test.mp4!");
    }

    if(Videofile!= NULL)
    {
        fwrite(pBuffer,iDataLen,1,Videofile); //下载视频文件
    }

    //调用播放库解码并显示码流实现回放
    /*
    if(1 == byDataType)
    {
        if (!PlayM4_GetPort(&m_lPort))
        {
            return FALSE;
        }
        if (!PlayM4_SetStreamOpenMode(m_lPort, STREAME_REALTIME))
        {
            return FALSE;
        }
        //Enter first 40-byte header
        if(!PlayM4_OpenStream(m_lPort, (unsigned char *)pBuffer,
(DWORD)iDataLen, 2*1024*1024))
        {
```

```
        return FALSE;
    }
    if(!PlayM4_Play(m_lPort, hWnd))
    {
        return FALSE;
    }
}
else
{
    for (int i=0; i<1000; i++)
    {
        BOOL bRet = PlayM4_InputData(m_lPort, (unsigned char *)pBuffer,
(DWORD)iDataLen);
        if (!bRet)
        {
            if ( i >=999)
            {
                printf("PlayM4_InputData failed, error code: %d!",
PlayM4_GetLastError(m_lPort));
            }
            Sleep(2);
        }
        else
        {
            break;
        }
    }
}
*/
return TRUE;
}

/////////////////////////////////////////////////////////////////
//注册回放码流回调函数
BOOL CALLBACK fnPLAYBACK_DATA_CB(LONG lPlayBackLinkHandle,
NET_EHOME_PLAYBACK_DATA_CB_INFO *pDataCBInfo, void *pUserData)
{
    if (NULL == pDataCBInfo)
    {
        return FALSE;
    }
    lPlayHandle = lPlayBackLinkHandle;
    InputStreamData (pDataCBInfo->dwType, (char*)pDataCBInfo->pData, pDataCBInfo->dwDataLen);
    return TRUE;
}

/////////////////////////////////////////////////////////////////
//注册回访请求的响应回调函数
BOOL CALLBACK fnPLAYBACK_NEWLINK_CB(LONG lPlayBackLinkHandle,
NET_EHOME_PLAYBACK_NEWLINK_CB_INFO *pNewLinkCBMsg, void *pUserData)
{
```

```
lLinkHandle = lPlayBackLinkHandle;
printf("Callback of playback listening, Device ID: %s\n", pNewLinkCBMsg->szDeviceID);

//回放数据的回调参数
NET_EHOME_PLAYBACK_DATA_CB_PARAM struDataCB = {0};
struDataCB.fnPlayBackDataCB = fnPLAYBACK_DATA_CB;
struDataCB.byStreamFormat = 0;//封装格式:0-PS 格式

if (!NET_ESTREAM_SetPlayBackDataCB(lPlayBackLinkHandle, &struDataCB))
{
    printf("NET_ESTREAM_SetPlayBackDataCB failed, error code: %d\n",
NET_ESTREAM_GetLastError());
    return FALSE;
}
printf("NET_ESTREAM_SetPlayBackDataCB!\n");

return TRUE;
}

////////////////////////////////////
////////////////////////////////////
//设置异常回调
NET_ESTREAM_SetExceptionCallBack(0, 0, StreamExceptionCallback, NULL);

void CALLBACK StreamExceptionCallback(DWORD dwType, LONG iUserID, LONG iHandle,
void* pUser)
{
    if(EHOME_PREVIEW_EXCEPTION == dwType)
    {
        //预览异常
        printf("Preview exception, handle=%d, Error:%d", iHandle, dwError);
    }
    else if (EHOME_PLAYBACK_EXCEPTION == dwType)
    {
        //回放异常
        printf("Playback exception, handle=%d, Error:%d", iHandle, dwError);
    }
    else if (EHOME_AUDIOTALK_EXCEPTION == dwType)
    {
        //语音对讲 (转发) 异常
        printf("Stream VoiceTalk exception, handle=%d, Error:%d", iHandle,
dwError);
    }
}

void main() {
    //////////////////////////////////////
    //SMS 在开启监听服务后获取码流

    //初始化 SMS 库
```

```
NET_ESTREAM_Init();

//回放的监听参数
NET_EHOME_PLAYBACK_LISTEN_PARAM struListen = {0};
memcpy(struListen.struIPAdress.szIP, "10.16.2.123", sizeof("10.16.2.123"));
struListen.struIPAdress.wPort = 8003; //SMS 的监听端口号
struListen.fnNewLinkCB = fnPLAYBACK_NEWLINK_CB; //回放请求的回调函数
struListen.pUserData = NULL;
struListen.byLinkMode = 0; //0-TCP, 1-UDP (保留)

//开启监听服务
LONG lHandle = NET_ESTREAM_StartListenPlayBack(&struListen);
if(lHandle < -1)
{
    printf("NET_ESTREAM_StartListenPlayBack failed, error code: %d\n",
NET_ESTREAM_GetLastError());
    NET_ESTREAM_Fini();
    return;
}
printf("NET_ESTREAM_StartListenPlayBack!\n");
////////////////////////////////////

////////////////////////////////////
//注册和回放请求

//初始化 CMS 库
NET_ECMS_Init();

//注册的监听参数
NET_EHOME_CMS_LISTEN_PARAM struCMSListenPara = {0};
memcpy(struCMSListenPara.struAddress.szIP, "0.0.0.0", sizeof("0.0.0.0"));
struCMSListenPara.struAddress.wPort = 7660;
struCMSListenPara.fnCB = RegisterCallBack;

//开启监听服务并接收设备注册信息
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
if(lListen < -1)
{
    printf("NET_ECMS_StartListen failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartListen!\n");

while(1)
{
    Sleep(1000); //以下操作需在注册完成后再进行
    if(lLoginID >= 0)
    {
        break;
    }
}
```

```
    }
}

//查找视频文件
NET_EHOME_REC_FILE_COND struFindCond = {0};
struFindCond.dwChannel = 1; //通道号, 从1开始
struFindCond.dwRecType = 0xff; //所有类型
struFindCond.dwStartIndex = 0; //查找起始位置
struFindCond.dwMaxFileCountPer = 5; //单次搜索的最多文件数量

//查找开始时间
struFindCond.struStartTime.wYear = (WORD)2015;
struFindCond.struStartTime.byMonth = (BYTE)5;
struFindCond.struStartTime.byDay = (BYTE)18;
struFindCond.struStartTime.byHour = (BYTE)10;
struFindCond.struStartTime.byMinute = (BYTE)0;
struFindCond.struStartTime.bySecond = (BYTE)0;

//查找结束时间
struFindCond.struStopTime.wYear = (WORD)2015;
struFindCond.struStopTime.byMonth = (BYTE)5;
struFindCond.struStopTime.byDay = (BYTE)18;
struFindCond.struStopTime.byHour = (BYTE)12;
struFindCond.struStopTime.byMinute = (BYTE)59;
struFindCond.struStopTime.bySecond = (BYTE)59;

LONG lSearchType = 0; //查找文件
LONG lFileHandle = NET_ECMS_StartFindFile_V11(lLoginID, lSearchType,
&struFindCond, sizeof(struFindCond)); //Start file search
if (lFileHandle < 0)
{
    printf("NET_ECMS_StartFindFile_V11 failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartFindFile_V11!\n");

LONG lRet = -1;
char csTmp[256] = {0};
char szFileName[MAX_FILE_NAME_LEN] = {0};

NET_EHOME_REC_FILE struFileInfo = {0};

//可在线程中搜索文件
while(1)
{
    lRet = NET_ECMS_FindNextFile_V11(lFileHandle, &struFileInfo,
sizeof(struFileInfo)); //逐个获取搜索结果
    if (lRet == ENUM_GET_NEXT_STATUS_SUCCESS)
    {
```

```
        if (struFileInfo.dwFileSize / 1024 == 0)
        {
            sprintf(csTmp, "%d", struFileInfo.dwFileSize);
        }
        else if (struFileInfo.dwFileSize / 1024 > 0 &&
struFileInfo.dwFileSize / (1024*1024) == 0)
        {
            sprintf(csTmp, "%dK", struFileInfo.dwFileSize/1024);
        }
        else
        {
            sprintf(csTmp, "%dM", struFileInfo.dwFileSize /1024/1024);
        }

        printf("Filename[%s], Filesize[%s], StarTime[%04d-%02d-%02d %02d:
%02d:%02d], StopTime[%04d-%02d-%02d %02d:%02d:%02d] \n", \
            struFileInfo.szFileName, csTmp,
struFileInfo.struStartTime.wYear, struFileInfo.struStartTime.byMonth, \
            struFileInfo.struStartTime.byDay,
struFileInfo.struStartTime.byHour, struFileInfo.struStartTime.byMinute, \
            struFileInfo.struStartTime.bySecond,
struFileInfo.struStopTime.byDay, struFileInfo.struStopTime.byHour, \
            struFileInfo.struStopTime.byMinute,
struFileInfo.struStopTime.bySecond); //Searched file information

        memcpy(szFileName, struFileInfo.sFileName, MAX_FILE_NAME_LEN);
    }
    else
    {
        if (lRet == ENUM_GET_NETX_STATUS_NEED_WAIT)
        {
            Sleep(5);
            continue;
        }

        if ((lRet == ENUM_GET_NETX_STATUS_NO_FILE) || (lRet ==
ENUM_GET_NEXT_STATUS_FINISH))
        {
            printf("No more file!\n");
            break;
        }
        else if(lRet == ENUM_GET_NEXT_STATUS_NOT_SUPPORT)
        {
            printf("Device does not support!\n");
            break;
        }
        else
        {
            printf("Failed to find a file, for the server is busy or
network failure!\n");
            break;
        }
    }
}
```

```

    }
}

//回放请求的输入参数
NET_EHOME_PLAYBACK_INFO_IN struPlayBackIn = {0};
struPlayBackIn.dwSize = sizeof(struPlayBackIn);
struPlayBackIn.dwChannel = 1; //通道号
struPlayBackIn.byPlayBackMode = 0; //回放模式：0-按文件名回放，1-按时间回放（保留）
struPlayBackIn.unionPlayBackMode.struPlayBackbyName.dwSeekType = 0; //0-按字节长度计算，1-按秒数计算
struPlayBackIn.unionPlayBackMode.struPlayBackbyName.dwFileOffset = 0;
//当 dwSeekType 为 0 时，偏移量按字节计算；当 dwSeekType 等于 1 时，偏移量按秒数计算
struPlayBackIn.unionPlayBackMode.struPlayBackbyName.dwFileSpan = 0; //已下载的文件大小，如果大小为 0，说明文件已下载完成
memcpy(struPlayBackIn.unionPlayBackMode.struPlayBackbyName.szFileName, szFileName, MAX_FILE_NAME_LEN); //需要进行回放的文件名称

memcpy(struPlayBackIn.struStreamSever.szIP, "10.16.2.123", sizeof("10.16.2.123")); //SMS 的 IP 地址
struPlayBackIn.struStreamSever.wPort = 8003; //SMS 的端口号，需和监听端口号一致

//回放请求的输出参数
NET_EHOME_PLAYBACK_INFO_OUT struPlayBackOut = {0};

//回放请求
if(!NET_ECMS_StartPlayBack(lLoginID, &struPlayBackIn, &struPlayBackOut))
{
    printf("NET_ECMS_StartPlayBack failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartPlayBack!\n");

//码流传输请求的输入参数
NET_EHOME_PUSHPLAYBACK_IN struPushPlayBackIn = {0};
struPushPlayBackIn.dwSize = sizeof(struPushPlayBackIn);
struPushPlayBackIn.lSessionID = struPlayBackOut.lSessionID; //回放请求的会话 ID

//码流传输请求的输出参数
NET_EHOME_PUSHPLAYBACK_OUT struPushPlayBackOut = {0};

//发送请求给设备并开始传输码流
if(!NET_ECMS_StartPushPlayBack(lLoginID, &struPushPlayBackIn, &struPushPlayBackOut))
{
    printf("NET_ECMS_StartPushPlayBack failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

```



```
}
printf("NET_ECMS_StartPushPlayBack!\n");

Sleep(50000);

////////////////////////////////////
//登出

//CMS 停止监听服务
if(!NET_ECMS_StopListen(lListen))
{
    printf("NET_ECMS_StopListen failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//释放被 CMS 占用的资源
NET_ECMS_Fini();
////////////////////////////////////

//SMS 停止转发码流
if(lPlayHandle >= 0)
{
    if (!NET_ESTREAM_StopPlayBack(lPlayHandle))
    {
        printf("NET_ESTREAM_StopPlayBack failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}

//SMS 停止监听服务
if(lLinkHandle >= 0)
{
    if (!NET_ESTREAM_StopListenPlayBack(lLinkHandle))
    {
        printf("NET_ESTREAM_StopListenPlayBack failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}

//释放被 SMS 占用的资源
NET_ESTREAM_Fini();

//释放文件资源
if(Videofile != NULL)
{
    fclose(Videofile);
    Videofile = NULL;
}

printf("Exit!\n");
}
```

后续处理

- 调用 `NET_ECMS_StopListen` 停止 CMS 监听服务。
- 调用 `NET_ECMS_Fini` 和 `NET_ESTREAM_Fini` 释放被 CMS 和 SMS 占用的资源。

第 6 章 语音对讲

6.1 开启语音对讲

语音对讲功能实现了监控中心和指定设备之间的音频数据发送和接收。它通常应用于以下场景：当某个公园或校园内存在多个出入口时，且每个出入口的岗亭均安装了相机，如果在某个出入口发生异常，那么监控中心人员可以通过语音对讲功能与对应岗亭内值班人员进行通话，了解具体情况从而发出处理或指挥命令。此处，语音对讲由中心管理服务（CMS）实现且该流程仅适用于支持 ISUP2.0 的设备。

前提条件

- 请确保已调用 `NET_ECMS_Init` 初始化 CMS。
- 请确保已调用 `NET_ECMS_StartListen` 开启 CMS 监听服务以便接收设备注册信息。

操作步骤

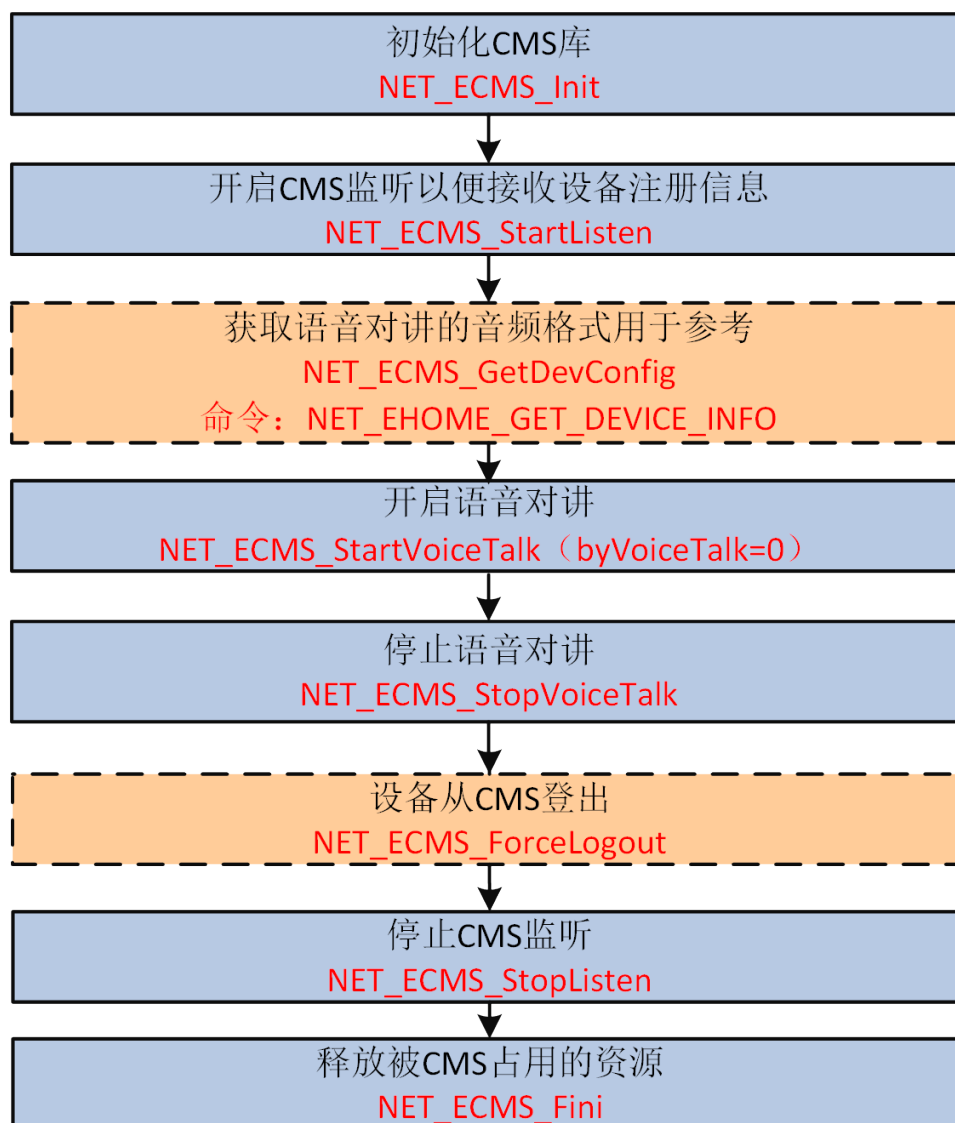


图 6-1 开启语音对讲的接口调用流程图

说明

为了判断设备支持的 ISUP 版本信息，可在调用 `NET_ECMS_StartListen` 开启 CMS 监听和注册回调函数时，将回调函数 `DEVICE_REGISTER_CB` 中的 `dwDataType` 设为“`ENUM_DEV_ON`”。设备支持的协议版本信息会通过结构体 `NET_EHOME_DEV_REG_INFO` 中的成员 `byDevProtocolVersion` 进行回调。

1. 可选操作: 调用 `NET_ECMS_GetDevConfig` 和“`NET_EHOME_GET_DEVICE_INFO`” (命令号: 1) 获取语音对讲支持的音频格式以便参考。

支持的音频格式由 **pOutBuf** 通过结构体 *NET_EHOME_DEVICE_INFO* 返回。

2. 调用 *NET_ECMS_StartVoiceTalk* 并将 *NET_EHOME_VOICETALK_PARA* 中的 **byVoiceTalk** 设置为“0”，从而开启语音对讲。
3. 调用 *NET_ECMS_StopVoiceTalk* 停止语音转发和对讲。
4. 可选操作: 调用 *NET_ECMS_ForceLogout* 强制注销设备。

示例

开启语音对讲的示例代码

```
#include <stdio.h>
#include <Windows.h>
#include "HCISUPCMS.h"

LONG lLoginID = -1;
LONG m_lVoiceTalkHandle = -1;

////////////////////////////////////
//注册回调函数
BOOL CALLBACK RegisterCallBack(LONG lUserID, DWORD dwDataType, void
*pOutBuffer, DWORD dwOutLen, void *pInBuffer, DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDevInfo = (NET_EHOME_DEV_REG_INFO *)pOutBuffer;

        if (pDevInfo != NULL)
        {
            lLoginID = lUserID;
            printf("On-line, lUserID: %d, Device ID: %s\n", lLoginID, pDevInfo->byDeviceID);
        }
        //输入参数
        NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO *)pInBuffer;
        pServerInfo->dwTimeOutCount = 6; //心跳超时次数
        pServerInfo->dwKeepAliveSec = 15; //心跳间隔
    }
    else if (ENUM_DEV_OFF == dwDataType)
    {
        printf("Off-line, lUserID: %d\n", lUserID);
        NET_ECMS_ForceLogout(lUserID);
    }
    else
    {
    }

    return TRUE;
}

////////////////////////////////////
//语音数据回调函数
```

```
void CALLBACK g_fVoiceDataCallBack(LONG iVoiceComHandle, char *pRecvDataBuffer,
DWORD dwBufSize, DWORD dwEncodeType, BYTE byAudioFlag, void *pUser)
{
    //获取到语音数据自行处理
}

void main() {

    ////////////////////////////////////////
    //CMS 注册和预览请求

    //CMS 注册模块初始化
    NET_ECMS_Init();

    //注册监听参数
    NET_EHOME_CMS_LISTEN_PARAM struCMSListenPara = {0};
    memcpy(struCMSListenPara.struAddress.szIP, "0.0.0.0", sizeof("0.0.0.0"));
    struCMSListenPara.struAddress.wPort = 7660;
    struCMSListenPara.fnCB = RegisterCallBack;

    //启动监听服务, 接收设备注册信息
    LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
    if(lListen < -1)
    {
        printf("NET_ECMS_StartListen failed, error code: %d\n",
NET_ECMS_GetLastError());
        NET_ECMS_Fini();
        return;
    }
    printf("NET_ECMS_StartListen!\n");

    while(1)
    {
        Sleep(1000); //注册成功之后再做后续操作
        if(lLoginID >= 0)
        {
            break;
        }
    }

    //语音对讲参数
    NET_EHOME_VOICETALK_PARA struVoiceTalkPara = {0};
    struVoiceTalkPara.bNeedCBNoEncData = 0; //需要回调的语音类型: 0- 编码后语音, 1- 编
码前语音(语音转发时不支持)
    struVoiceTalkPara.cbVoiceDataCallBack = g_fVoiceDataCallBack;
    struVoiceTalkPara.dwEncodeType = 0; //设备的语音编码类型: 0- G722, 1- G711U, 2-
G711A, 3- G726, 4- AAC, 5- MP2L2, 6- PCM
    struVoiceTalkPara.byVoiceTalk = 0; //语音对讲模式: 0- 语音对讲, 1- 语音转发

    DWORD dwVoiceChan = 1; //语音对讲通道号
```

```

//开始语音对讲
m_lVoiceTalkHandle = NET_ECMS_StartVoiceTalk(lLoginID, dwVoiceChan,
&struVoiceTalkPara);
if (m_lVoiceTalkHandle < 0)
{
    printf("NET_ECMS_StartVoiceTalk failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

Sleep(30000);

//停止语音对讲
if(!NET_ECMS_StopVoiceTalk(m_lVoiceTalkHandle))
{
    printf("NET_ECMS_StopVoiceTalk failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

////////////////////////////////////
//退出

//CMS 停止监听服务
if(!NET_ECMS_StopListen(lListen))
{
    printf("NET_ECMS_StopListen failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//CMS 反初始化, 释放资源
NET_ECMS_Fini();

printf("Exit!\n");
}

```

后续处理

- 调用 `NET_ECMS_StopListen` 停止 CMS 监听服务。
- 调用 `NET_ECMS_Fini` 释放被 CMS 占用的资源。

6.2 语音转发

若某公园或者校园内存在多个出入口, 且各个出入口对应的岗亭都已安装了摄像机, 那么监控中心可以根据需要通过语音转发功能将音频信息发送给某个指定摄像机进行对话。这里, 语音转发功能通过中心管理服务 (CMS) 实现。

前提条件

- 请确保已调用 `NET_ECMS_Init` 初始化 CMS。
- 请确保已调用 `NET_ECMS_StartListen` 开启 CMS 监听服务以便接收设备注册信息。

操作步骤

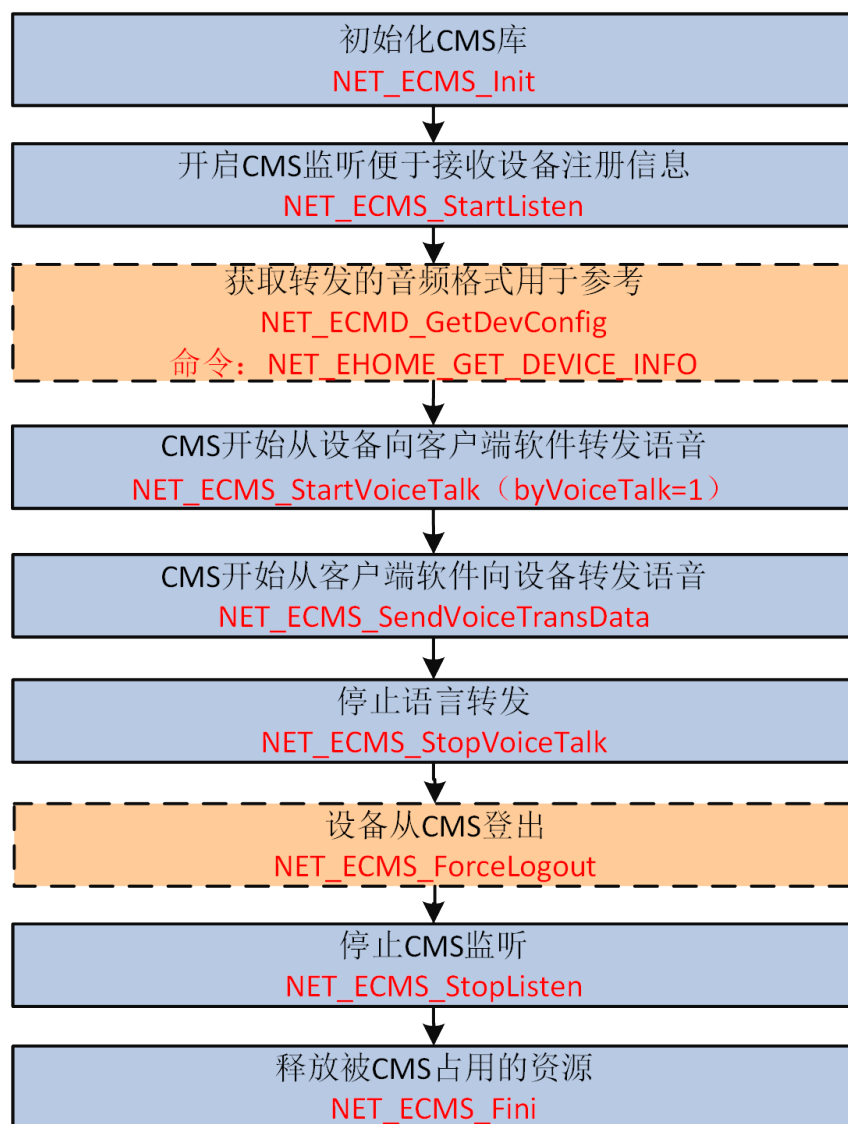


图 6-2 开始语音转发的接口调用流程图

1. 可选操作: 调用 `NET_ECMS_GetDevConfig` 和 “NET_EHOME_GET_DEVICE_INFO” (命令号: 1) 获取语音对讲支持的音频格式以便参考。
支持的音频格式由 `pOutBuf` 通过结构体 `NET_EHOME_DEVICE_INFO` 返回。
2. 调用 `NET_ECMS_StartVoiceTalk` 并将 `NET_EHOME_VOICETALK_PARA` 中的 `byVoiceTalk` 设置为 “1”，从而将来自设备的语音转发给客户端软件。

3. 调用 `NET_ECMS_SendVoiceTransData` 将来自客户端软件的语音转发给设备。
4. 调用 `NET_ECMS_StopVoiceTalk` 停止语音转发和对讲。
5. 可选操作: 调用 `NET_ECMS_ForceLogout` 强制注销设备。

后续处理

- 调用 `NET_ECMS_StopListen` 停止 CMS 监听服务。
- 调用 `NET_ECMS_Fini` 释放被 CMS 占用的资源。

6.3 通过流媒体服务开启语音对讲

语音对讲功能实现了监控中心和指定设备之间的音频数据发送和接收。它通常应用于以下场景：当某个公园或校园内存在多个出入口时，且每个出入口的岗亭均安装了相机，如果在某个出入口发生异常，那么监控中心人员可以通过语音对讲功能与对应岗亭内值班人员进行通话，了解具体情况从而发出处理或指挥命令。此处，语音对讲由流媒体服务（SMS）和中心管理服务（CMS）实现且该流程适用于支持 ISUP4.0 和 ISUP5.0 的设备。

前提条件

- 请确保已调用 `NET_ECMS_Init` 和 `NET_ESTREAM_Init` 初始化 CMS 和 SMS。
- 请确保已调用 `NET_ECMS_StartListen` 开启 CMS 监听服务以便接收设备注册信息。

操作步骤

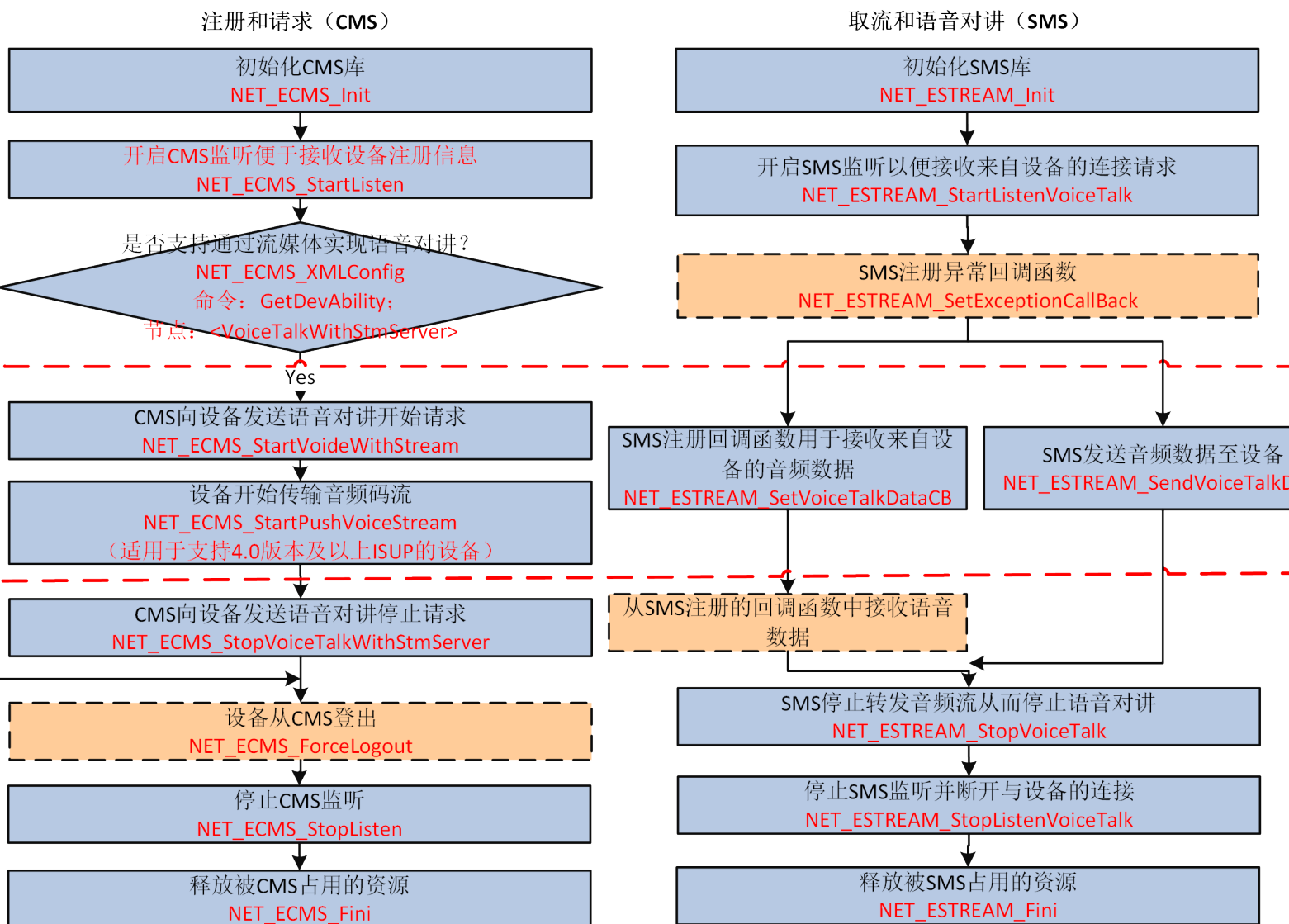


图 6-3 通过流媒体服务开启语音对讲的接口调用流程图

说明

为了判断设备支持的 ISUP 版本信息, 可在调用 `NET_ECMS_StartListen` 开启 CMS 监听和注册回调函数时, 将回调函数 `DEVICE_REGISTER_CB` 中的 `dwDataType` 设为“`ENUM_DEV_ON`”。设备支持的协议版本信息会通过结构体 `NET_EHOME_DEV_REG_INFO` 中的成员 `byDevProtocolVersion` 进行回调。

1. 调用 `NET_ECMS_XMLConfig` 和命令 `GetDevAbility` 获取设备能力并判断设备是否支持通过流媒体服务开启语音对讲。

设备能力集由 **pOutBuf** 返回。

如果支持，将返回节点<**VoiceTalkWithStmServer**>，可继续下面步骤。

否则，请结束该任务。

2. 调用 **NET_ESTREAM_StartListenPreview** 开启 SMS 监听服务并接收来自设备的连接请求。

3. 可选操作: 调用 **NET_ESTREAM_SetExceptionCallBack** 为 SMS 注册异常回调函数。

4. 调用 **NET_ECMS_StartVoiceWithStmServer** 将语音对讲开启请求从 CMS 发送给设备。

发送 SMS 的地址和端口号给设备，设备自动为 CMS 分配一个会话 ID。

5. 调用 **NET_ECMS_StartPushVoiceStream** 将语音传输请求从 CMS 发送给设备。

设备自动连接 SMS 并开始发送音频数据给 SMS

6. 执行下面某个操作接收来自设备的音频数据或发送音频数据至设备。

- a. 调用 **NET_ESTREAM_SetVoiceTalkDataCB** 为 SMS 注册回调函数用于接收来自设备的音频数据。

b. 从 SMS 注册的语音回调函数中获取音频数据并将其转发给客户端软件。

说明

语音对讲音频码流的解码和显示由客户端软件实现。

- 调用 **NET_ESTREAM_SendVoiceTalkData** 将音频数据发送给设备。

7. 调用 **NET_ESTREAM_StopVoiceTalk** 停止 SMS 的音频码流转发。

8. 调用 **NET_ECMS_StopVoiceTalkWithStmServer** 将语音对讲停止请求从 CMS 发送给设备。

设备停止传输音频码流。

9. 调用 **NET_ESTREAM_StopListenVoiceTalk** 停止 SMS 监听服务并断开其与设备的连接。

10. 可选操作: 调用 **NET_ECMS_ForceLogout** 强制设备登出 CMS。

示例

通过流媒体服务开启语音对讲的示例代码

说明

在该示例代码中，CMS 和 SMS 被安装在同一台电脑的同一个程序中。

```
#include <stdio.h>
#include <Windows.h>
#include "HCEHomeCMS.h"
#include "HCEHomeStream.h"
#include "plaympeg4.h"

LONG lLoginID = -1;
LONG lVoiceLinkHandle = -1;
LONG lVoiceHandle = -1;
FILE AudioTalkfile = NULL;

////////////////////////////////////
```

```
//注册回调函数
BOOL CALLBACK RegisterCallBack(LONG lUserID, DWORD dwDataType, void
*pOutBuffer, DWORD dwOutLen, void *pInBuffer, DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDevInfo = (NET_EHOME_DEV_REG_INFO *)pOutBuffer;

        if (pDevInfo != NULL)
        {
            lLoginID = lUserID;
            printf("On-line, lUserID: %d, Device ID: %s\n", lLoginID, pDevInfo-
>byDeviceID);
        }
        //输入参数
        NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO *)pInBuffer;
        pServerInfo->dwTimeOutCount = 6; //心跳超时次数
        pServerInfo->dwKeepAliveSec = 15; //心跳间隔
    }
    else if (ENUM_DEV_OFF == dwDataType)
    {
        printf("Off-line, lUserID: %d\n", lUserID);
        NET_ECMS_ForceLogout(lUserID);
    }
    else
    {
    }

    return TRUE;
}

////////////////////////////////////
//语音数据处理
BOOL InputVoiceData(char* pBuffer, int iDataLen)
{
    if(AudioTalkfile == NULL)
    {
        AudioTalkfile = fopen("Test.mp4","wb");
        printf("Save data to file: Test.mp4!\n");
    }

    if(AudioTalkfile!= NULL)
    {
        fwrite(pBuffer,iDataLen,1,AudioTalkfile); //回调实时流直接写文件, 保存录像
    }

    //.....其他数据处理, 语音数据转发、解码播放需要自行实现

    return TRUE;
}
```

```
////////////////////////////////////
//语音数据回调函数
void CALLBACK fnVOICETALK_DATA_CB(LONG lHandle,
NET_EHOME_VOICETALK_DATA_CB_INFO *pDataCBInfo, void *pUserData)
{
    if (NULL == pDataCBInfo)
    {
        return ;
    }
    lVoiceHandle = lHandle;
    InputVoiceData((char*)pDataCBInfo->pData, pDataCBInfo->dwDataLen);
}

////////////////////////////////////
//SMS 语音对讲请求回应回调函数
BOOL CALLBACK fnVOICETALK_NEWLINK_CB(LONG lLinkHandle,
NET_EHOME_VOICETALK_NEWLINK_CB_INFO *pNewLinkCBInfo, void *pUserData)
{
    lVoiceLinkHandle = lLinkHandle;
    printf("Callback of two-way audio listening, Device ID: %s, Audio Channel:
%d, Audio Type: %d\n", pNewLinkCBInfo->szDeviceID, pNewLinkCBInfo->dwAudioChan,
pNewLinkCBInfo->dwEncodeType);

    //语音数据回调参数
    NET_EHOME_VOICETALK_DATA_CB_PARAM struDataCB = {0};
    struDataCB.fnVoiceTalkDataCB = fnVOICETALK_DATA_CB;

    if (!NET_ESTREAM_SetVoiceTalkDataCB(lVoiceLinkHandle, &struDataCB))
    {
        printf("NET_ESTREAM_SetVoiceTalkDataCB failed, error code: %d\n",
NET_ESTREAM_GetLastError());
        return FALSE;
    }
    printf("NET_ESTREAM_SetVoiceTalkDataCB!\n");

    return TRUE;
}

////////////////////////////////////
//读取文件语音数据发给设备
DWORD WINAPI TestVoiceTransThread(LPVOID lpArg)
{
    LONG lHanle = (LONG)lpArg;

    FILE *pFile = fopen("ehome_send.audio", "rb"); //打开本地的语音文件, 音频格式需要
跟设备语音对讲音频格式一致
    if (pFile == NULL)
    {
        return 0;
    }
}
```

```
char szBuf[80] = {0}; //以 G722 音频类型为例, 每次发送数据 80 字节
int nRemainLen = 0;

int nCurPos = fseek(pFile,0, SEEK_END);
nRemainLen = ftell(pFile);
fseek(pFile, nCurPos, SEEK_SET);

int nSendTotal = 0;
while(1)
{
    if (nRemainLen >= 80)
    {
        nSendTotal += 80;
        fread(szBuf, 80, 1, pFile);

        语音对讲数据参数
        NET_EHOME_VOICETALK_DATA struVoicTalkData = {0};
        struVoicTalkData.pSendBuf = (BYTE*)szBuf;
        struVoicTalkData.dwDataLen = 80;

        NET_ESTREAM_SendVoiceTalkData(lHanle, &struVoicTalkData);
        nRemainLen -= 80;
    }
    else
    {
        nSendTotal += nRemainLen;
        fread(szBuf, nRemainLen, 1, pFile);

        语音对讲数据参数
        NET_EHOME_VOICETALK_DATA struVoicTalkData = {0};
        struVoicTalkData.pSendBuf = (BYTE*)szBuf;
        struVoicTalkData.dwDataLen = nRemainLen;

        NET_ESTREAM_SendVoiceTalkData(lHanle, &struVoicTalkData);
        break;
    }
    Sleep(15);
}

fclose(pFile);
return 0;
}

void main(){
    //////////////////////////////////////
    //SMS 开启服务

    //SMS 初始化
    NET_ESTREAM_Init();

    //语音对讲监听参数
```

```
NET_EHOME_LISTEN_VOICETALK_CFG struListen = {0};
memcpy(struListen.struIPAdress.szIP, "10.16.2.123", sizeof("10.16.2.123"));
struListen.struIPAdress.wPort = 8003; //流媒体服务监听端口
struListen.fnNewLinkCB = fnVOICETALK_NEWLINK_CB; //语音对讲连接请求回调函数
struListen.pUser = NULL;

//启动语音对讲监听
LONG lHandle = NET_ESTREAM_StartListenVoiceTalk(&struListen);
if(lHandle < -1)
{
    printf("NET_ESTREAM_StartListenVoiceTalk failed, error code: %d\n",
NET_ESTREAM_GetLastError());
    NET_ESTREAM_Fini();
    return;
}
printf("NET_ESTREAM_StartListenVoiceTalk!\n");
////////////////////////////////////

////////////////////////////////////
//CMS 注册和语音对讲请求

//CMS 注册模块初始化
NET_ECMS_Init();

//注册监听参数
NET_EHOME_CMS_LISTEN_PARAM struCMSListenPara = {0};
memcpy(struCMSListenPara.struAddress.szIP, "0.0.0.0", sizeof("0.0.0.0"));
struCMSListenPara.struAddress.wPort = 7660;
struCMSListenPara.fnCB = RegisterCallBack;

//启动监听服务, 接收设备注册信息
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
if(lListen < -1)
{
    printf("NET_ECMS_StartListen failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartListen!\n");

while(1)
{
    Sleep(1000); //注册成功之后再做后续操作
    if(lLoginID >= 0)
    {
        break;
    }
}

//语音对讲请求输入参数
```

```
NET_EHOME_VOICE_TALK_IN struVoiceTalkIn = {0};
struVoiceTalkIn.dwVoiceChan = 1; //语音对讲通道号
memcpy(struVoiceTalkIn.struStreamSever.szIP, "10.16.2.123",
sizeof("10.16.2.123")); //流媒体服务 IP 地址
struVoiceTalkIn.struStreamSever.wPort = 8003; //流媒体服务端口, 需要跟服务监听端口
一致

//语音对讲请求输出参数
NET_EHOME_VOICE_TALK_OUT struVoiceTalkOut = {0};

//CMS 语音对讲请求
if(!NET_ECMS_StartVoiceWithStmServer(lLoginID, &struVoiceTalkIn,
&struVoiceTalkOut))
{
    printf("NET_ECMS_StartVoiceWithStmServer failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartVoiceWithStmServer!\n");

//CMS 请求开始推流
NET_EHOME_PUSHVOICE_IN struPushVoiceIn = {0};
struPushVoiceIn.dwSize = sizeof(struPushVoiceIn);
struPushVoiceIn.lSessionID = struVoiceTalkOut.lSessionID;
NET_EHOME_PUSHVOICE_OUT struPushVoiceOut = {0};
struPushVoiceOut.dwSize = sizeof(struPushVoiceOut);
if (!NET_ECMS_StartPushVoiceStream(lLoginID, &struPushVoiceIn,
&struPushVoiceOut))
{
    printf("NET_ECMS_StartPushVoiceStream failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartPushVoiceStream!\n");

//创建线程从文件读取语音数据发送给设备
DWORD dwThreadId;
CreateThread(NULL, 0, LPTHREAD_START_ROUTINE(TestVoiceTransThread),
(void*)lVoiceLinkHandle, 0, &dwThreadId);

Sleep(50000);

////////////////////////////////////
//退出

//释放 CMS 语音对讲请求资源
if(!NET_ECMS_StopVoiceTalkWithStmServer(lLoginID,
struVoiceTalkOut.lSessionID))
{
```



```
    printf("NET_ECMS_StopVoiceTalkWithStmServer failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//CMS 停止监听服务
if(!NET_ECMS_StopListen(lListen))
{
    printf("NET_ECMS_StopListen failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//CMS 反初始化, 释放资源
NET_ECMS_Fini();
////////////////////////////////////

//SMS 停止语音对讲
if(lVoiceHandle >= 0)
{
    if (!NET_ESTREAM_StopVoiceTalk(lVoiceHandle))
    {
        printf("NET_ESTREAM_StopVoiceTalk failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}

//SMS 停止语音对讲监听服务
if(lVoiceLinkHandle >= 0)
{
    if (!NET_ESTREAM_StopListenVoiceTalk(lVoiceLinkHandle))
    {
        printf("NET_ESTREAM_StopListenVoiceTalk failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}

//SMS 反初始化, 释放资源
NET_ESTREAM_Fini();

//释放文件资源
if(AudioTalkfile != NULL)
{
    fclose(AudioTalkfile);
    AudioTalkfile = NULL;
}

printf("Exit!\n");
}
```

后续处理

- 调用 `NET_ECMS_StopListen` 停止 CMS 监听服务。
- 调用 `NET_ECMS_Fini` 和 `NET_ESTREAM_Fini` 释放被 CMS 和 SMS 占用的资源。

第 7 章 云台控制

云台控制实现了相机的水平和上下转动以及缩放功能。可通过为指定监控区域设置预置点从而按需快速切换相机画面或通过 3D 定位重新将选择的区域定位至画面中心。

控制云台	调用 <i>NET_ECMS_RemoteControl</i> ，将 dwCommand 设置为“NET_EHOME_PTZ_CTRL”（命令号：1000），并将 <i>NET_EHOME_REMOTE_CTRL_PARAM</i> 中的 lplnbuffer 设置为 <i>NET_EHOME_PTZ_PARAM</i> 。
配置、调用和清除预置点	调用 <i>NET_ECMS_RemoteControl</i> ，将 dwCommand 设置为“NET_EHOME_PRESET_CTRL”（命令号：1001），并将 <i>NET_EHOME_REMOTE_CTRL_PARAM</i> 中的 lplnbuffer 设置为 <i>NET_EHOME_PRESET_PARAM</i> 。
执行 3D 定位	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>PZIN</i> 。

第 8 章 报警或事件配置

该章节显示了支持通过 ISUPSDK 配置的报警或事件类型。在触发和接收报警或事件之前，可对报警或事件参数，如规则、布防计划和报警联动行为进行设置。

说明

如果配置了报警或事件参数，可启用报警监听服务，以便在报警触发或事件发生时接收设备自动上传的报警或事件信息（详见 [开启监听服务用于接收报警](#)）。有关报警或事件详情，请参见 [表 16-1](#) 和 [事件类型和详情](#)。

移动侦测

表 8-1 移动侦测

获取移动侦测参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetMotionPara 。
配置移动侦测参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetMotionPara 。
获取移动侦测区域参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetMotionArea 。
配置移动侦测区域参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetMotionArea 。

视频遮挡侦测

表 8-2 视频遮挡侦测

获取视频遮挡侦测参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetHideAlarmPara 。
配置视频遮挡侦测参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetHideAlarmPara 。
获取视频遮挡侦测区域参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetHideAlarmArea 。
配置视频遮挡侦测区域参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetHideAlarmArea 。

视频丢失侦测

表 8-3 视频丢失侦测

获取视频丢失侦测参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetVILostPara</u> 。
设置视频丢失侦测参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetVILostPara</u> 。

隐私遮蔽

表 8-4 隐私遮蔽

获取隐私遮蔽参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetHidePara</u> 。
配置隐私遮蔽参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetHidePara</u> 。
获取隐私遮蔽区域参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetHideArea</u> 。
配置隐私遮蔽区域参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetHideArea</u> 。

异常报警

表 8-5 异常报警

获取异常报警参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetExceptionPara</u> 。
配置异常报警参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetExceptionPara</u> 。

客流统计

表 8-6 客流统计

获取客流统计参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetPassengerPara</u> 。
设置客流统计参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetPassengerPara</u> 。
获取客流统计信息 OSD 参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetPassengerOSDPara</u> 。
配置客流统计信息 OSD 参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetPassengerOSDPara</u> 。

获取客流统计检测线参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>GetPassengerLinePara</i> 。
配置客流统计检测线参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>SetPassengerLinePara</i> 。
获取客流统计摄像机标定参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>GetCalibrationPara</i> 。
配置客流统计摄像机标定参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>SetCalibrationPara</i> 。

报警输入

表 8-7 报警输入

获取报警输入参数	调用 <i>NET_ECMS_GetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_GET_ALARMIN_CFG”（命令号：11），并将 pCondBuf 设置为 <i>NET_EHOME_ALARMIN_COND</i> 。 报警输入参数由 pOutBuf 通过结构体 <i>NET_EHOME_ALARMIN_CFG</i> 返回。
获取报警输出参数	调用 <i>NET_ECMS_SetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_SET_ALARMIN_CFG”（命令号：12），将 pCondBuf 设置为 <i>NET_EHOME_ALARMIN_COND</i> ，并将 pInBuf 设置为 <i>NET_EHOME_ALARMIN_CFG</i> 。

报警布防计划

表 8-8 报警布防计划

获取报警布防计划参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>GetAlarmDeploymentTime</i> 。
配置报警布防计划参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>SetAlarmDeploymentTime</i> 。

报警联动

表 8-9 报警联动

获取报警联动参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>GetAlarmLinkageType</i> 。
配置报警联动参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>SetAlarmLinkageType</i> 。
获取抓拍联动参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>GetAlarmTriggerCapture</i> 。
配置抓拍联动参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>SetAlarmTriggerCapture</i> 。
获取录像联动参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>GetAlarmTriggerRecord</i> 。
配置录像联动参数	调用 <i>NET_ECMS_XMLConfig</i> 传输命令报文 <i>SetAlarmTriggerRecord</i> 。
手动控制报警输出联动	调用 <i>NET_ECMS_RemoteControl</i> ，将 dwCommand 设置为“NET_EHOME_MANUAL_IOOUT”（命令号：19），并将 <i>NET_EHOME_REMOTE_CTRL_PARAM</i> 中的 lplnbuffer 设置为 <i>NET_EHOME_MANUAL_IOOUT_CTRL</i> 。
获取报警输出参数	调用 <i>NET_ECMS_GetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_GET_ALARMOUT_CFG”（命令号：15），并将 pCondBuf 设置为 4 位报警输入号。 报警输出参数由 pOutBuf 通过结构体 <i>NET_EHOME_ALARMOUT_CFG</i> 返回。
配置报警输出参数	调用 <i>NET_ECMS_SetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_SET_ALARMOUT_CFG”（命令号：16），将 pCondBuf 设置为 4 位报警输入号，并将 pInBuf 设置为 <i>NET_EHOME_ALARMOUT_CFG</i> 。

第 9 章 开启监听服务用于接收报警

通过 ISUPSDK 开启监听服务并接收设备报警信息依赖于中心管理服务（CMS）和报警管理服务（AMS）。监听服务开启后，当报警被触发时，设备会自动上传报警，因此可通过配置监听服务（即 AMS）接收报警信息。

前提条件

- 请确保已调用 `NET_ECMS_Init` 和 `NET_EALARM_Init` 初始化 CMS 和 AMS。
- 请确保已调用 `NET_ECMS_StartListen` 开启 CMS 监听以接收设备注册信息。

操作步骤

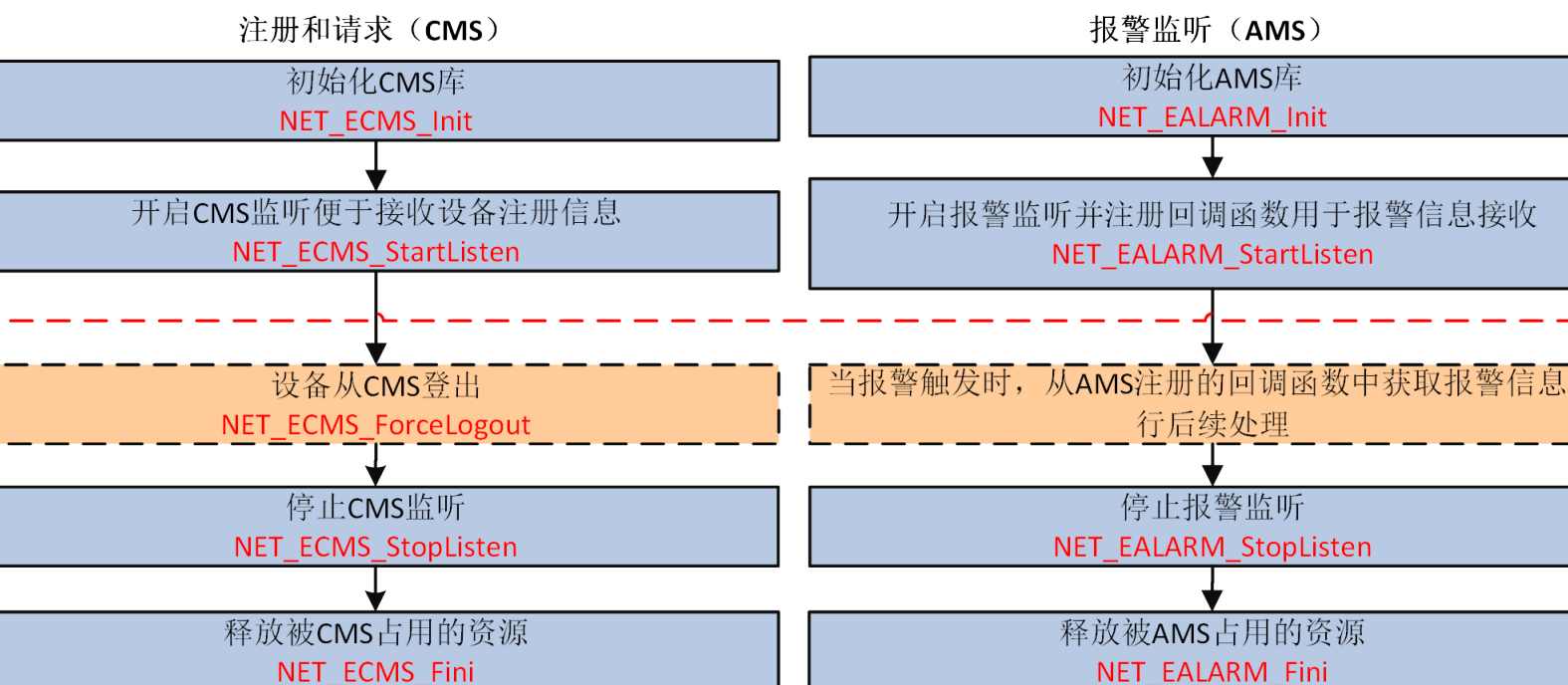


图 9-1 开启监听服务用于接收报警的接口调用流程图

1. 调用 `NET_EALARM_StartListen` 开启 AMS 报警监听并注册回调函数用于接收报警信息。

AMS 的 IP 地址和端口号由 CMS 发送给设备。

2. 可选操作: 为支持 ISUP5.0 的设备设置 SessionKey。

说明

SessionKey 通过回调数据类型“enum_dev_sessionkey”获得，详见 [设备注册消息处理流程](#)。

3. 当报警触发时，从 AMS 注册的回调函数中获取报警信息进行处理。

4. 调用 `NET_EALARM_StopListen` 停止 AMS 报警监听。

5. 可选操作: 调用 `NET_ECMS_ForceLogout` 强制注销设备。

示例

开启监听服务用于接收报警的示例代码

说明

在该示例代码中, CMS 和 AMS 安装在同一台电脑的同一个程序中。

```
#include <stdio.h>
#include <Windows.h>
#include "HCISUPCMS.h"
#include "HCISUPAlarm.h"

LONG lLoginID = -1;
LONG lListenHandle = -1;

////////////////////////////////////
//注册回调函数
BOOL CALLBACK RegisterCallBack(LONG lUserID, DWORD dwDataType, void
*pOutBuffer, DWORD dwOutLen, void *pInBuffer, DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDevInfo = (NET_EHOME_DEV_REG_INFO *)pOutBuffer;

        if (pDevInfo != NULL)
        {
            lLoginID = lUserID;
            printf("On-line, lUserID: %d, Device ID: %s\n", lLoginID, pDevInfo-
>byDeviceID);
        }
        //输入参数
        NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO *)pInBuffer;
        pServerInfo->dwTimeOutCount = 6; //心跳超时次数
        pServerInfo->dwKeepAliveSec = 15; //心跳间隔
        memcpy(pServerInfo->struUDPAlarmSever.szIP, "10.16.2.123",
sizeof("10.16.2.123")); //报警服务器 IP 地址 (TCP 协议)
        pServerInfo->struUDPAlarmSever.wPort = 7200; //报警服务器端口 (UDP 协议), 需要
和报警服务器启动监听的端口一致
        pServerInfo->dwAlarmServerType = 0; //报警服务器类型: 0- 只支持 UDP 协议上报,
1- 支持 UDP、TCP 两种协议上报
    }
    else if (ENUM_DEV_OFF == dwDataType)
    {
        printf("Off-line, lUserID: %d\n", lUserID);
        NET_ECMS_ForceLogout(lUserID);
    }
    else
```



```
{
}

return TRUE;
}

////////////////////////////////////
//AMS 报警回调函数
BOOL CALLBACK AlarmMSGCallBack(LONG lHandle, NET_EHOME_ALARM_MSG *pAlarmMsg,
void *pUserData)
{
    lListenHandle = lHandle;

    DWORD dwType = pAlarmMsg->dwAlarmType; //不同的报警类型(dwAlarmType),
pAlarmInfo 对应不同的报警信息类型
    printf("Callback of alarm listening, dwAlarmType[%d]\n", dwType);

    switch(dwType)
    {
    case EHOME_ALARM_UNKNOWN://未知报警信息
        printf("Unknown Alarm Type!\n");
        break;
    case EHOME_ALARM://基本报警信息:移动侦测、视频遮盖、视频丢失、PIR 报警、人脸侦测、区域入侵等
        NET_EHOME_ALARM_INFO struAlarmInfo;
        memcpy(&struAlarmInfo, pAlarmMsg->pAlarmInfo,
sizeof(NET_EHOME_ALARM_INFO));
        printf("Basic Alarm: Device ID[%s], szAlarmTime[%s], Alarm Type[%d],
Action[%d], VideoChannel[%d]!\n", struAlarmInfo.szDeviceID, \
            struAlarmInfo.szAlarmTime, struAlarmInfo.dwAlarmType,
struAlarmInfo.dwAlarmAction, struAlarmInfo.dwVideoChannel);
        break;
    case EHOME_ALARM_HEATMAP_REPORT://热度图报告上传
        NET_EHOME_HEATMAP_REPORT struHeatmapInfo;
        memcpy(&struHeatmapInfo, pAlarmMsg->pAlarmInfo,
sizeof(NET_EHOME_HEATMAP_REPORT));
        printf("Heatmap report: Device ID[%s], StartTime[%s], StopTime[%s],
dwVideoChannel[%d]!\n", struHeatmapInfo.byDeviceID, \
            struHeatmapInfo.byStartTime, struHeatmapInfo.byStopTime,
struHeatmapInfo.dwVideoChannel);
        break;
    case EHOME_ALARM_FACESNAP_REPORT://人脸抓拍报告上传
        NET_EHOME_FACESNAP_REPORT struFaceSnap;
        memcpy(&struFaceSnap, pAlarmMsg->pAlarmInfo,
sizeof(NET_EHOME_FACESNAP_REPORT));
        printf("Face snap: Device ID[%s], AlarmTime[%s], dwFaceScore[%d],
dwVideoChannel[%d]!\n", struFaceSnap.byDeviceID, \
            struFaceSnap.byAlarmTime, struFaceSnap.dwFaceScore,
struFaceSnap.dwVideoChannel);
        break;
    case EHOME_ALARM_GPS://GPS 信息上传
        NET_EHOME_GPS_INFO struGpsInfo;
```

```
        memcpy(&struGpsInfo, pAlarmMsg->pAlarmInfo, sizeof(NET_EHOME_GPS_INFO));
        printf("GPS info: Device ID[%s], SampleTime[%s], Longitude[%d],
Latitude[%d]!\n", struGpsInfo.byDeviceID, \
                struGpsInfo.bySampleTime, struGpsInfo.dwLongitude,
struGpsInfo.dwLatitude );
        break;
    case EHOME_ALARM_CID_REPORT://报警主机 CID 报警信息
        NET_EHOME_CID_INFO struCIDInfo;
        memcpy(&struCIDInfo, pAlarmMsg->pAlarmInfo, sizeof(NET_EHOME_CID_INFO));
        printf("CID info: Device ID[%s], AlarmTime[%s], CID Code[%d], CID
Type[%d], CID Describe[%s]!\n", struCIDInfo.byDeviceID, \
                struCIDInfo.byTriggerTime, struCIDInfo.dwCIDCode,
struCIDInfo.dwCIDType, struCIDInfo.byCIDDescribe );
        break;
    default:
        break;
}

return TRUE;
}

void main(){

////////////////////////////////////
//开启 AMS 报警监听

//AMS 初始化
NET_EALARM_Init();

//报警监听参数
NET_EHOME_ALARM_LISTEN_PARAM struListen = {0};
memcpy(struListen.struAddress.szIP, "10.16.2.123", sizeof("10.16.2.123"));
struListen.struAddress.wPort = 7200; //报警服务的监听端口
struListen.fnMsgCb = AlarmMSGCallBack; //报警回调函数
struListen.pUserData = NULL;
struListen.byProtocolType = 1; //0- TCP 方式(保留, 暂不支持), 1- UDP 方式

//启动报警监听
LONG lHandle = NET_EALARM_StartListen(&struListen);
if(lHandle < -1)
{
    printf("NET_EALARM_StartListen failed, error code: %d\n",
NET_EALARM_GetLastError());
    NET_EALARM_Fini();
    return;
}
printf("NET_EALARM_StartListen!\n");
////////////////////////////////////

////////////////////////////////////
//CMS 注册模块初始化
```

```
NET_ECMS_Init();

//注册监听参数
NET_EHOME_CMS_LISTEN_PARAM struCMSListenPara = {0};
memcpy(struCMSListenPara.struAddress.szIP, "0.0.0.0", sizeof("0.0.0.0"));
struCMSListenPara.struAddress.wPort = 7660;
struCMSListenPara.fnCB = RegisterCallBack;

//启动监听, 接收设备注册信息, 注册回调函数里面需要发送报警主机 IP 和端口给设备
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
if(lListen < -1)
{
    printf("NET_ECMS_StartListen failed, error code: %d\n",
NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}
printf("NET_ECMS_StartListen!\n");

while(1)
{
    Sleep(1000); //注册成功之后再后续操作
    if(lLoginID >= 0)
    {
        break;
    }
}

Sleep(300000); //等待 5 分钟, 接收报警信息

////////////////////////////////////
//退出

//CMS 停止监听
if(!NET_ECMS_StopListen(lListen))
{
    printf("NET_ECMS_StopListen failed, error code: %d\n",
NET_ECMS_GetLastError());
}

//CMS 反初始化, 释放资源
NET_ECMS_Fini();

//AMS 停止报警监听
if(lListenHandle >= 0)
{
    if (!NET_EALARM_StopListen(lListenHandle))
    {
        printf("NET_EALARM_StopListen failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}
```

```
}  
//AMS 反初始化, 释放资源  
NET_EALARM_Fini();  
  
printf("Exit!");  
}
```

后续处理

- 调用 *NET_ECMS_StopListen* 停止 CMS 监听。
- 调用 *NET_ECMS_Fini* 和 *NET_EALARM_Fini* 释放被 CMS 和 AMS 占用的资源。

第 10 章 设备维护

ISUPSDK 提供了获取设备信息（包括版本信息）和设备工作状态的接口和命令。也可通过该 SDK 重启、升级设备或手动校时。

获取设备信息	调用 <i>NET_ECMS_GetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_GET_DEVICE_CFG”（命令号：3）。 设备信息由 pOutBuf 通过结构体 <i>NET_EHOME_DEVICE_CFG</i> 返回。
配置设备信息	调用 <i>NET_ECMS_SetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_SET_DEVICE_CFG”（命令号：4），并将 pInBuf 设置为 <i>NET_EHOME_DEVICE_CFG</i> 。
查询设备工作状态	<i>GETDEVICEWORKSTATUS</i>
重启设备	<i>REBOOT</i>
升级设备	<i>UPDATE</i>
执行手动校时	<i>ADJUSTTIME</i>
获取设备版本信息	调用 <i>NET_ECMS_GetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_GET_VERSION_INFO”（命令号：2）。 设备版本信息由 pOutBuf 通过结构体 <i>NET_EHOME_DEVICE_CFG</i> 返回。

第 11 章 存储管理

通过 ISUP SDK 集成存储管理功能依赖于存储管理服务 (SS)。SS 支持保存文件，且支持多种协议类型，如 Tomcat 协议，VRB 协议，和 ISUP5.0。

下面两张示意图展示了 SS 在文件上传和下发应用中扮演的角色。

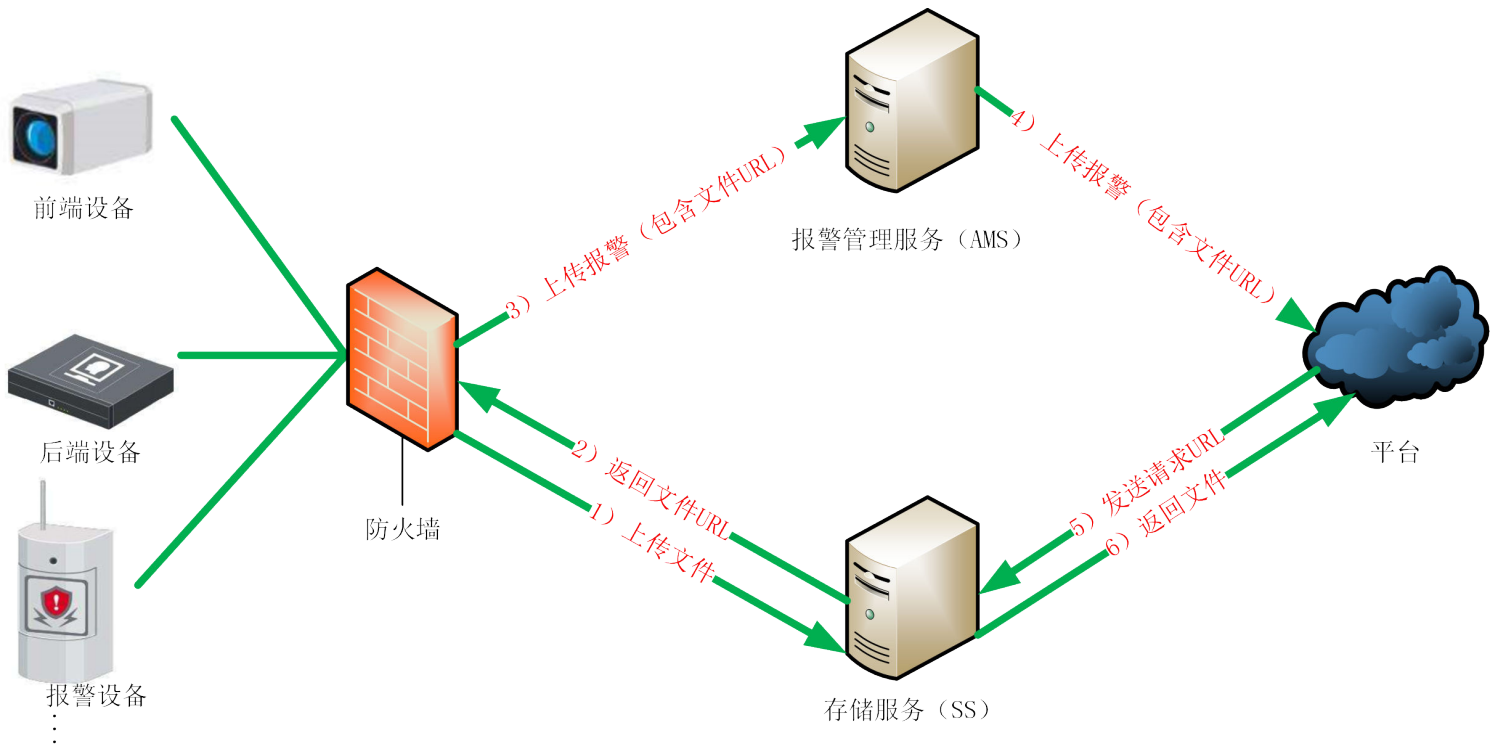


图 11-1 设备上传文件至平台

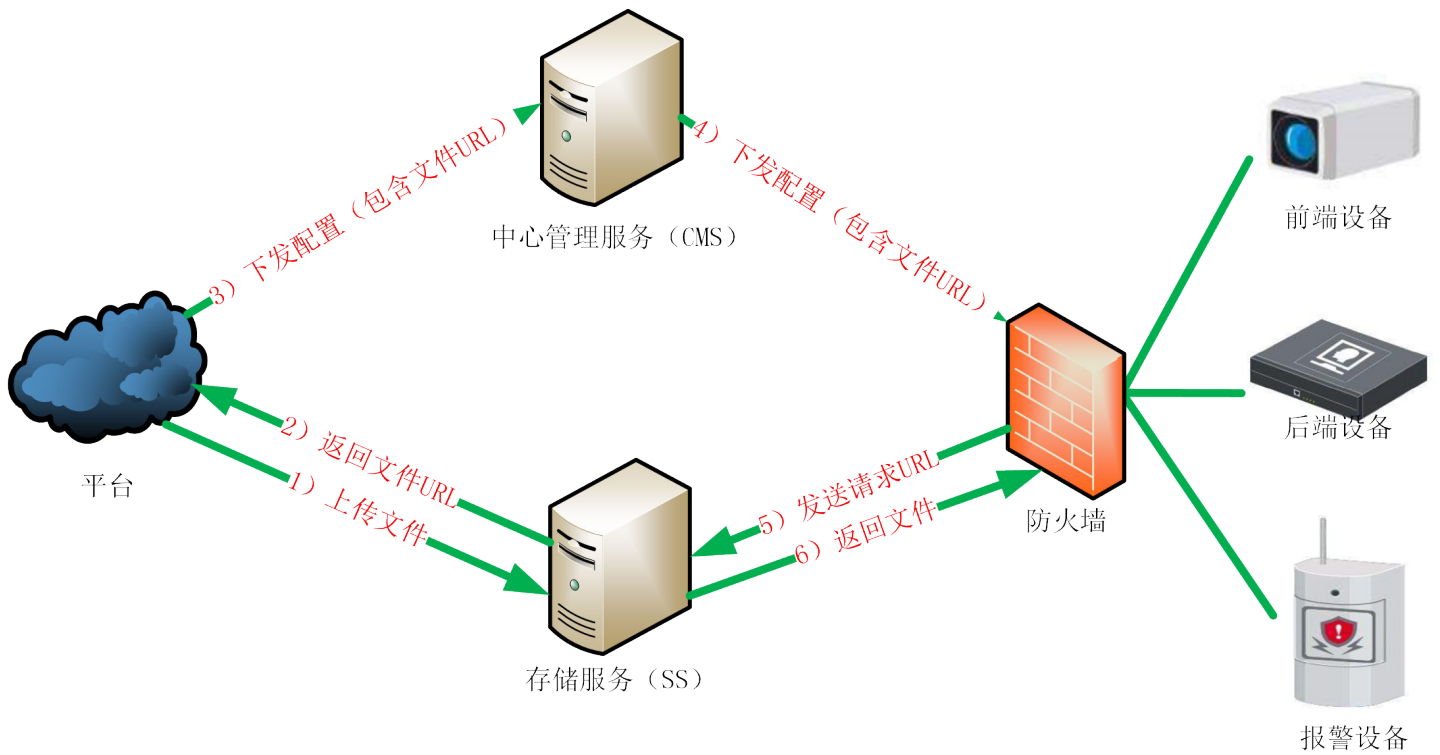


图 11-2 平台下发文件至设备

11.1 开启存储服务

在开始上传文件至存储服务 (SS) 或从存储服务 (SS) 下载文件之前，必须要开启存储服务。此外支持开启安全模式。

操作步骤

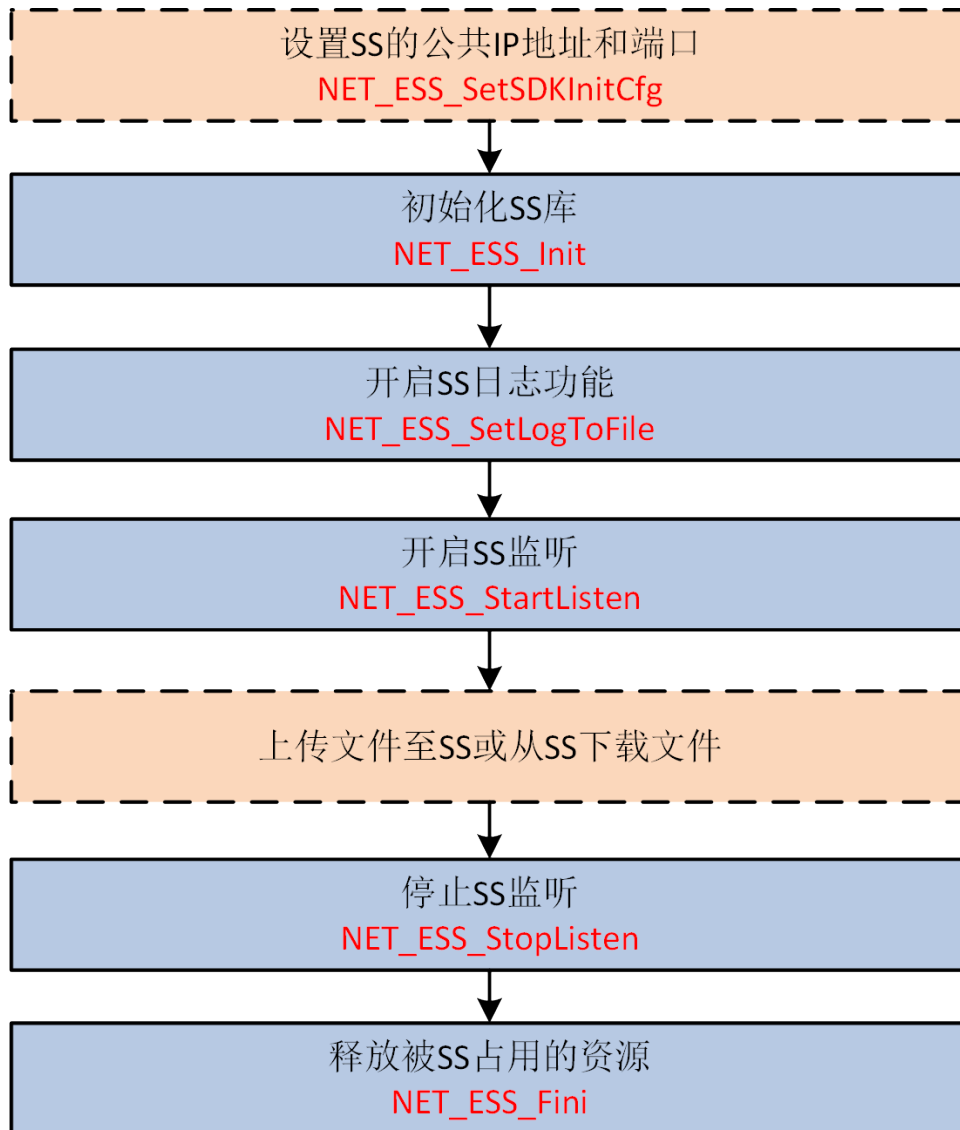


图 11-3 开启存储服务的接口调用流程图

1. 可选操作: 调用 `NET_ESS_SetSDKInitCfg`，设置 `enumType` 为“`NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT`”，并将 `IpInBuff` 设置为 `NET_EHOME_IPADDRESS`，从而设置 SS 的公共 IP 地址和端口。

 说明

当 SS 在公共网络环境下时，该步骤为必选。

2. 调用 `NET_ESS_Init` 初始化 SS。
3. 调用 `NET_ESS_SetLogToFile` 为 SS 开启日志功能。
4. 调用 `NET_ESS_StartListen` 开启 SS 监听。

 说明

通过此接口可开启安全模式，在该模式下，文件必须通过认证后才可进行上传、下载或删除操作。

5. 可选操作: 上传文件至 SS 或从 SS 下载文件。

6. 调用 `NET_ESS_StopListen` 停止 SS 监听。

示例

开启存储服务的示例代码

```
#include <stdio.h>
#include <iostream>
#include "../incCn/HCISUPSS.h"

#define SS_STORAGE_PATH "C:\\Picture" //文件保存路径
#define PIC_URI_LEN 128 //图片 URL 大小

//信息回调函数
int SS_Message_Callback(LONG iHandle, NET_EHOME_SS_MSG_TYPE enumType
    , void *pOutBuffer, DWORD dwOutLen, void *pInBuffer, DWORD dwInLen, void
    *pUser)
{
    if (enumType == NET_EHOME_SS_MSG_TOMCAT)
        { //回调 Tomcat 协议信息

            LPNET_EHOME_SS_TOMCAT_MSG pTomcatMsg =
(LPNET_EHOME_SS_TOMCAT_MSG)pOutBuffer;
            char szPicUri[PIC_URI_LEN * 4] = { 0 };
            for (int i = 0; i < pTomcatMsg->dwPicNum; i++)
                {
                    sprintf(szPicUri + i * PIC_URI_LEN, "%s"
                        , pTomcatMsg->pPicURLs + (i * MAX_URL_LEN_SS));
                }
            char szUrlHead[PIC_URI_LEN] = { 0 };
            memcpy(szUrlHead, pTomcatMsg->szDevUri, 128);
            int picNum = pTomcatMsg->dwPicNum;
            char szMsg[5 * PIC_URI_LEN] = { 0 };
            sprintf(szMsg, "tomcat:uri[%s],picNum:[%d] ,picInfo:[%s]"
                , szUrlHead, picNum, szPicUri);
            printf(szMsg);

            FILE* pFile = fopen("C:/Picture/tomcatOutput.txt", "w+");
            if (pFile != NULL)
                {
                    for (int i = 0; i < pTomcatMsg->dwPicNum; i++)
                        {
                            fwrite(szPicUri + i * PIC_URI_LEN, 1, strlen(szPicUri + i *
PIC_URI_LEN), pFile);
                            fwrite("\n", 1, 1, pFile);
                        }
                    fclose(pFile);
                }
        }
}
```

```
    }
}
else if (enumType == NET_EHOME_SS_MSG_KMS_USER_PWD)
{
    *(BOOL*)pInBuffer = 1;
}
else if (enumType == NET_EHOME_SS_MSG_CLOUD_AK)
{
    strncpy((char*)pInBuffer, "5e998dbbafb44ca783099afcdead40fa7A3Vf7Fh",
dwInLen);
}
return TRUE;
}

//用于保存文件的回调函数
int SS_Storage_Callback(LONG iHandle, const char* pFileName, void *pFileBuf,
DWORD dwFileLen, char *pFilePath, void *pUser)
{
    if (pFileName == NULL || pFileBuf == NULL || dwFileLen == 0)
    {
        return FALSE;
    }

    //判断文件保存路径是否已存在, 若不存在, 创建一个文件
    if (!PathIsDirectory(SS_STORAGE_PATH))
    {
        if (!CreateDirectory(SS_STORAGE_PATH, NULL))
        {
            return FALSE;
        }
    }

    //保存文件
    char strFilePath[MAX_PATH] = { 0 };
    sprintf(strFilePath, "%s\\%s", SS_STORAGE_PATH, pFileName);

    FILE* pFile = fopen(strFilePath, "wb+");
    if (pFile == NULL)
    {
        return FALSE;
    }

    uint dwWriteLen = fwrite(pFileBuf, 1, dwFileLen, pFile);

    fclose(pFile);
    if (dwWriteLen != dwFileLen)
    {
        return FALSE;
    }

    strncpy(pFilePath, strFilePath, 259);
    return TRUE;
}
```

```
}

int main()
{
    //初始化资源
    NET_ESS_Init();
    //开启日志功能
    NET_ESS_SetLogToFile(3, "C:/SdkLog/", TRUE);

    //开启监听
    NET_EHOME_SS_LISTEN_PARAM struSSLListenParam = { 0 };
    memcpy(struSSLListenParam.struAddress.szIP, "10.8.97.60",
strlen("10.8.97.60"));
    struSSLListenParam.struAddress.wPort = 8080;
    memcpy(struSSLListenParam.szKMS_UserName, "test", strlen("test"));
memcpy(struSSLListenParam.szKMS_Password, "12345", strlen("12345"));
    memcpy(struSSLListenParam.szAccessKey, "test", strlen("test"));
memcpy(struSSLListenParam.szSecretKey, "12345", strlen("12345"));
    struSSLListenParam.fnSMsgCb = SS_Message_Callback;
    struSSLListenParam.fnSStorageCb = SS_Storage_Callback;
    struSSLListenParam.pUserData = NULL;
    long m_lSSHandle = NET_ESS_StartListen(&struSSLListenParam);
    if (-1 == m_lSSHandle)
    {
        printf("NET_ESS_StartListen Failed, port : %d",
struSSLListenParam.struAddress.wPort);
    }
    else
    {
        printf("NET_ESS_StartListen succ port : %d",
struSSLListenParam.struAddress.wPort);
    }

    char cTmp = '\\0';
    do
    {
        printf("Input q to exit!");
        cTmp = getchar();
    }while(cTmp != 'q');

    //停止监听
    NET_ESS_StopListen(m_lSSHandle);
    //释放资源
    NET_ESS_Fini();

    return 0;
}
```

后续处理

调用 `NET_ESS_Fini` 释放由 SS 占用的资源。

11.2 上传文件至存储服务

在 ISUPSDK 集成过程中，在平台和设备之间传输的文件格式必须是 URL，可通过 URL 下载文件信息。因此在上传文件至平台或下发文件至设备前，设备或平台必须将文件上传至存储服务（SS）以便获取文件 URL。此外，支持安全模式下上传或删除文件。

前提条件

请确保已开启 SS，详见 [开启存储服务](#)。

操作步骤

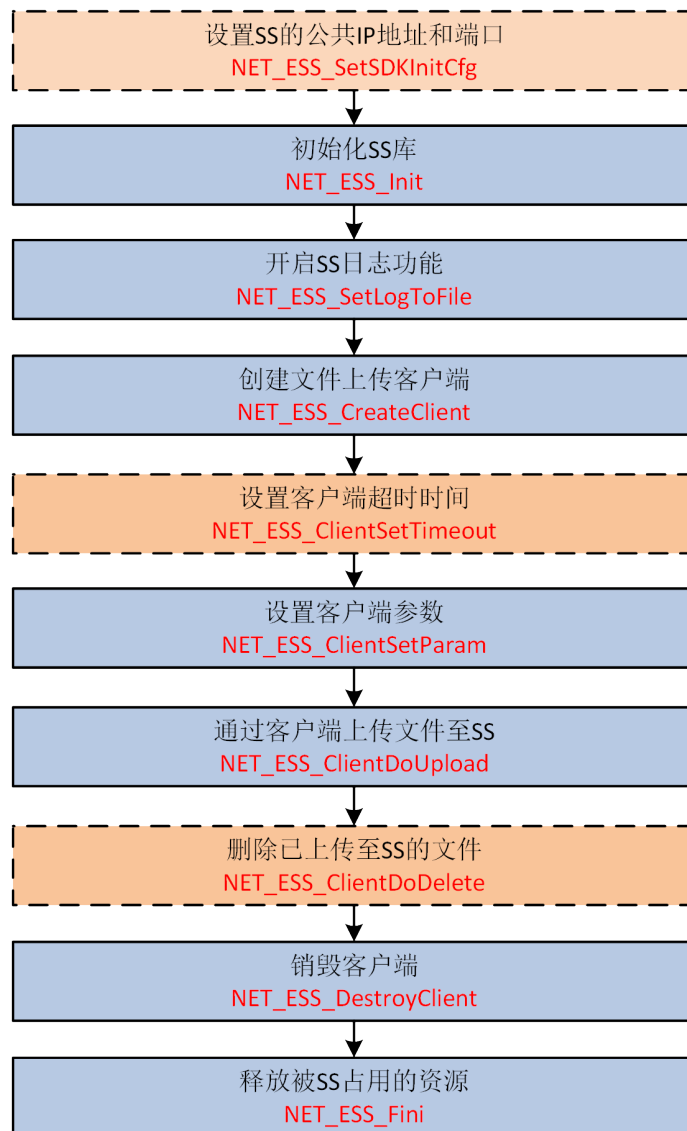


图 11-4 上传文件至存储服务的接口调用流程图

1. 可选操作: 调用 `NET_ESS_SetSDKInitCfg`，设置 `enumType` 为“`NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT`”，并将 `IpInBuff` 设置为 `NET_EHOME_IPADDRESS`，从而设置 SS 的公共 IP 地址和端口。

说明

当 SS 在公共网络环境下时，该步骤为必选。

2. 调用 `NET_ESS_Init` 初始化 SS。
3. 调用 `NET_ESS_SetLogToFile` 为 SS 开启日志功能。
4. 调用 `NET_ESS_CreateClient` 创建文件上传客户端。

说明

在安全模式下，客户端类型不能为“`NET_EHOME_SS_CLIENT_TYPE_TOMCAT`”（支持 Tomcat 协议的客户端）和“`NET_EHOME_SS_CLIENT_TYPE_VRB`”（支持 VRB 协议的客户端）。

5. 可选操作: 调用 `NET_ESS_ClientSetTimeout` 设置上传客户端的超时时间。
6. 调用 `NET_ESS_ClientSetParam` 设置上传客户端参数。

说明

在安全模式下，文件的认证方式包括密钥管理服务（KMS）和云存储服务，即需要为 KMS 设置用户名和密码，或为云存储服务设置 `AccessKey` 和 `SecretKey`。

7. 调用 `NET_ESS_ClientDoUpload` 通过客户端上传文件至 SS。

上传的文件将被保存在 SS 中，此外对应的文件 URL 会返回给设备。

8. 可选操作: 调用 `NET_ESS_ClientDoDelete` 删除已上传至 SS 的文件。
9. 完成上传之后，调用 `NET_ESS_DestroyClient` 销毁上传客户端。

示例

（非安全模式）上传文件至存储服务的示例代码

```
#include <stdio.h>
#include <iostream>
#include "../incCn/HCISUPSS.h"

int main()
{
    //初始化 SS
    NET_ESS_Init();
    //开启日志功能
    NET_ESS_SetLogToFile(3, "C:/SdkLog/", TRUE);

    NET_EHOME_SS_CLIENT_PARAM struClientParam;
    memset(&struClientParam, 0, sizeof(struClientParam));
    struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_TOMCAT;
    memcpy(struClientParam.struAddress.szIP, "10.8.97.60",
    strlen("10.8.97.60"));
    struClientParam.struAddress.wPort = 8080;
```

```
//基于 Tomcat 协议上传文件至 ss
lSSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (lSSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(lSSClientHandle, 60 * 1000, 60 * 1000);
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_FILE_PATH_PARAM_NAME,
"C:\\Picture\\1.jpg");

    char szUrl[MAX_URL_LEN_SS] = { 0 };
    NET_ESS_ClientDoUpload(lSSClientHandle, szUrl, MAX_URL_LEN_SS - 1);

    NET_ESS_DestroyClient(lSSClientHandle);
}

//基于 VRB 协议上传文件至 ss
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_VRB;
lSSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (lSSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(lSSClientHandle, 60 * 1000, 60 * 1000);

    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_FILE_PATH_PARAM_NAME,
"C:\\Picture\\1.jpg");
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_VRB_FILENAME_CODE,
"filename=1.jpg&code=");

    NET_ESS_ClientDoUpload(lSSClientHandle, szUrl, MAX_URL_LEN_SS - 1);

    NET_ESS_DestroyClient(lSSClientHandle);
}

//基于 KMS 协议上传文件至 ss
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_KMS;
lSSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (lSSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(lSSClientHandle, 60 * 1000, 60 * 1000);

    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_FILE_PATH_PARAM_NAME,
"C:\\Picture\\1.jpg");
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_KMS_USER_NAME,
"test");
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_KMS_PASSWIRD,
"12345");

    NET_ESS_ClientDoUpload(lSSClientHandle, szUrl, MAX_URL_LEN_SS - 1);

    DWORD dwErr = NET_ESS_GetLastError();

    NET_ESS_DestroyClient(lSSClientHandle);
}
```

```
//基于 ISUP5.0 上传文件至 SS
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_CLOUD;
lSSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (lSSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(lSSClientHandle, 60 * 1000, 60 * 1000);

    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_FILE_PATH_PARAM_NAME,
"C:\\\\Picture\\\\1.jpg");
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_CLOUD_AK_NAME,
"test");
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_CLOUD_SK_NAME,
"12345");

    NET_ESS_ClientDoUpload(lSSClientHandle, szUrl, MAX_URL_LEN_SS - 1);

    DWORD dwErr = NET_ESS_GetLastError();

    byte* bFileContent = NULL;

    NET_ESS_DestroyClient(lSSClientHandle);
}

//释放被 SS 占用的资源
NET_ESS_Fini();

return 0;
}
```

后续处理

调用 `NET_ESS_Fini` 释放由 SS 占用的资源。

11.3 从存储服务下载文件

在 ISUPSDK 集成过程中，由于文件以 URL 格式进行传输且文件详情保存在存储服务中，因此用户需要根据文件 URL 从存储服务中下载文件。此外，支持安全模式下下载文件。

前提条件

请确保已开启 SS，详见 [开启存储服务](#)。

操作步骤

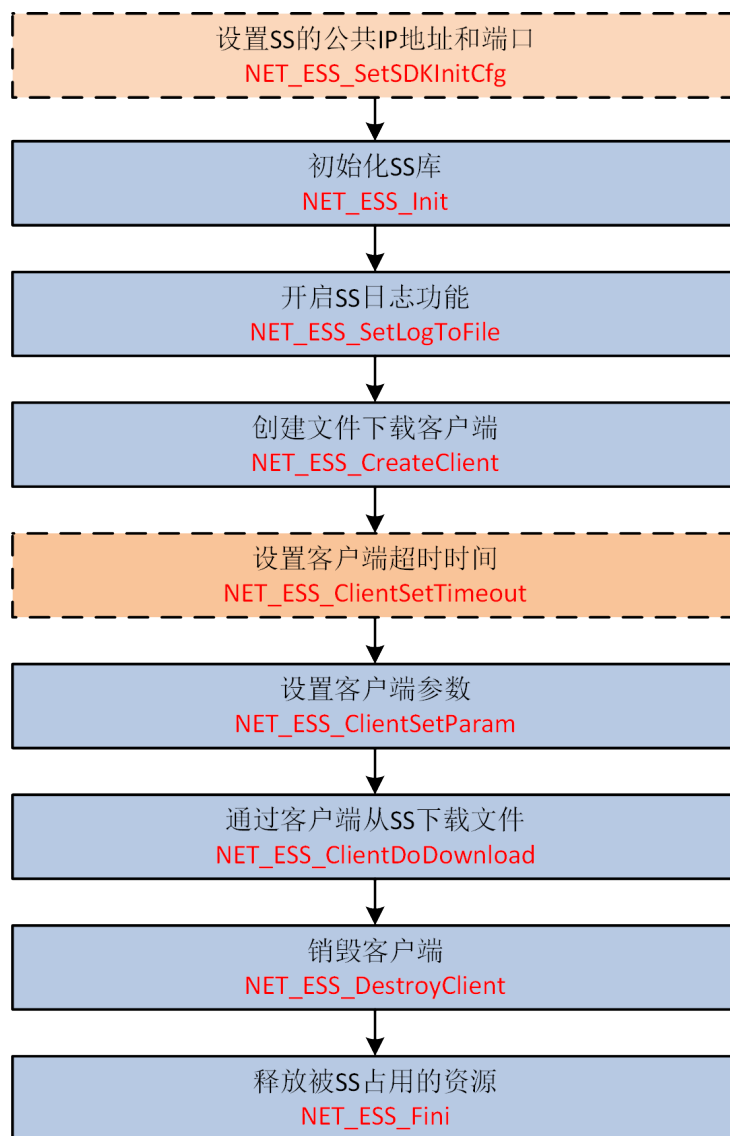


图 11-5 从存储服务下载文件的接口调用流程图

1. 可选操作: 调用 `NET_ESS_SetSDKInitCfg`，设置 `enumType` 为“`NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT`”，并将 `IpInBuff` 设置为 `NET_EHOME_IPADDRESS`，从而设置 SS 的公共 IP 地址和端口。

 说明

当 SS 在公共网络环境下时，该步骤为必选。

2. 调用 `NET_ESS_Init` 初始化 SS。
3. 调用 `NET_ESS_SetLogToFile` 为 SS 开启日志功能。
4. 调用 `NET_ESS_CreateClient` 创建文件下载客户端。

说明

在安全模式下，客户端类型不能为“NET_EHOME_SS_CLIENT_TYPE_TOMCAT”（支持 Tomcat 协议的客户端）和“NET_EHOME_SS_CLIENT_TYPE_VRB”（支持 VRB 协议的客户端）。

5. 可选操作: 调用 `NET_ESS_ClientSetTimeout` 设置下载客户端的超时时间。
6. 调用 `NET_ESS_ClientSetParam` 设置下载客户端参数。

说明

在安全模式下，文件的认证方式包括密钥管理服务（KMS）和云存储服务，即需要为 KMS 设置用户名和密码，或为云存储服务设置 `AccessKey` 和 `SecretKey`。

7. 调用 `NET_ESS_ClientDoDownload` 根据文件 URL 并通过下载客户端从 SS 中下载文件。
8. 完成下载之后，调用 `NET_ESS_DestroyClient` 销毁下载客户端。

示例

（非安全模式）从存储服务器下载文件的示例代码

```
#include <stdio.h>
#include <iostream>
#include "../incCn/HCSUPSS.h"

int main()
{
    //初始化 SS
    NET_ESS_Init();
    //开启日志功能
    NET_ESS_SetLogToFile(3, "C:/SdkLog/", TRUE);

    NET_EHOME_SS_CLIENT_PARAM struClientParam;
    memset(&struClientParam, 0, sizeof(struClientParam));
    struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_TOMCAT;
    memcpy(struClientParam.struAddress.szIP, "10.8.97.60",
    strlen("10.8.97.60"));
    struClientParam.struAddress.wPort = 8080;

    //基于 Tomcat 协议和 VRB 协议从 SS 下载文件
    struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_VRB;
    lSSClientHandle = NET_ESS_CreateClient(&struClientParam);
    if (lSSClientHandle >= 0)
    {
        NET_ESS_ClientSetTimeout(lSSClientHandle, 60 * 1000, 60 * 1000);

        byte* bFileContent = NULL;
        DWORD dwFileLen = 0;
        if (NET_ESS_ClientDoDownload(lSSClientHandle, szUrl,
        (void**) &bFileContent, dwFileLen))
        {
            FILE* pFile = fopen("C://Picture//vr.b.jpg", "wb+");
            if (pFile != NULL)
```

```
        {
            fwrite(bFileContent, 1, dwFileLen, pFile);
            fclose(pFile);
        }
    }

    //销毁下载客户端
    NET_ESS_DestroyClient(lSSClientHandle);
}

//基于 KMS 协议从 SS 下载文件
lSSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (lSSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(lSSClientHandle, 60 * 1000, 60 * 1000);

    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_KMS_USER_NAME,
"test");
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_KMS_PASSWIRD,
"12345");

    byte* bFileContent = NULL;
    DWORD dwFileLen = 0;
    if (NET_ESS_ClientDoDownload(lSSClientHandle, szUrl,
(void**) &bFileContent, dwFileLen))
    {
        FILE* pFile = fopen("C://Picture//kms.jpg", "wb+");
        if (pFile != NULL)
        {
            fwrite(bFileContent, 1, dwFileLen, pFile);
            fclose(pFile);
        }
    }

    DWORD dwErr = NET_ESS_GetLastError();

    //销毁下载客户端
    NET_ESS_DestroyClient(lSSClientHandle);
}

//基于 5.0 版本 ISUP 从 SS 下载文件
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_CLOUD;
lSSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (lSSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(lSSClientHandle, 60 * 1000, 60 * 1000);

    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_CLOUD_AK_NAME,
"test");
    NET_ESS_ClientSetParam(lSSClientHandle, SS_CLIENT_CLOUD_SK_NAME,
"12345");
}
```

```
byte* bFileContent = NULL;
DWORD dwFileLen = 0;
if (NET_ESS_ClientDoDownload(lSSClientHandle, szUrl,
(void**) &bFileContent, dwFileLen))
{
    FILE* pFile = fopen("C://Picture//cloud.jpg", "wb+");
    if (pFile != NULL)
    {
        fwrite(bFileContent, 1, dwFileLen, pFile);
        fclose(pFile);
    }
}

NET_ESS_DestroyClient(lSSClientHandle);
}

//释放被 SS 占用的资源
NET_ESS_Fini();

return 0;
}
```

后续处理

调用 `NET_ESS_Fini` 释放由 SS 占用的资源。

第 12 章 参数配置

该章节显示了 ISUPSDK 支持的基础或高级参数配置，如网络、编码、图像、录像和串口等。

表 12-1 网络配置

获取网络参数	调用 <i>NET_ECMS_GetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_GET_NETWORK_CFG”（命令号：5）。 网络参数由 pOutBuf 通过结构体 <i>NET_EHOME_NETWORK_CFG</i> 返回。
配置网络参数	调用 <i>NET_ECMS_SetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_SET_NETWORK_CFG”（命令号：6），并将 pInBuf 设置为 <i>NET_EHOME_NETWORK_CFG</i> 。

表 12-2 编码配置

获取编码参数	调用 <i>NET_ECMS_GetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_GET_COMPRESSION_CFG”（命令号：7），并将 pCondBuf 设置为 <i>NET_EHOME_COMPRESSION_COND</i> 。 编码参数由 pOutBuf 通过结构体 <i>NET_EHOME_COMPRESSION_CFG</i> 返回。
配置编码参数	调用 <i>NET_ECMS_SetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_SET_COMPRESSION_CFG”（命令号：8），将 pCondBuf 设置为 <i>NET_EHOME_COMPRESSION_COND</i> ，并将 pInBuf 设置为 <i>NET_EHOME_NETWORK_CFG</i> 。

表 12-3 图像和 OSD 配置

获取 OSD 参数	调用 <i>NET_ECMS_GetDevConfig</i> ，将 dwCommand 设置为“NET_EHOME_GET_PIC_CFG”（命令号：23）。
-----------	---

	OSD 参数由 pOutBuf 通过结构体 <u>NET_EHOME_PIC_CFG</u> 返回。
配置 OSD 参数	调用 <u>NET_ECMS_SetDevConfig</u> ，将 dwCommand 设置为“NET_EHOME_SET_PIC_CFG”（命令号：24），并将 pInBuf 设置为 <u>NET_EHOME_PIC_CFG</u> 。
获取图像参数	调用 <u>NET_ECMS_GetDevConfig</u> ，将 dwCommand 设置为“NET_EHOME_GET_IMAGE_CFG”（命令号：9），并将 pCondBuf 设置为 4 位通道号。 图像参数由 pOutBuf 通过结构体 <u>NET_EHOME_IMAGE_CFG</u> 返回。
配置图像参数	调用 <u>NET_ECMS_SetDevConfig</u> ，将 dwCommand 设置为“NET_EHOME_SET_IMAGE_CFG”（命令号：10），将 pCondBuf 设置为 4 位通道号，并将 pInBuf 设置为 <u>NET_EHOME_IMAGE_CFG</u> 。

表 12-4 录像配置

获取录像参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetRecordPlanPara</u> 。
设置录像参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetRecordPlanPara</u> 。

表 12-5 串口配置

获取 RS-485 串口参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetRS485Para</u> 。
配置 RS-485 串口参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetRS485Para</u> 。
获取 RS-232 串口参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>GetRS232Para</u> 。
配置 RS-232 串口参数	调用 <u>NET_ECMS_XMLConfig</u> 传输命令报文 <u>SetRS232Para</u> 。

表 12-6 其他配置

远程布防和撤防	调用 <i>NET_ECMS_XMLRemoteControl</i> 传输命令报文 <i>SETDEFENSESTATUS</i> 。
重启算法库	调用 <i>NET_ECMS_XMLRemoteControl</i> 传输命令报文 <i>RESTARTIVSLIB</i> 。

12.1 配置功耗模式

开启设备的休眠唤醒模式后，在相同的电量下，设备使用时间更长。当设备处于休眠模式时，预览、回放、语音对讲、配置等任何信令接口都无效，只有唤醒设备后，正常的信令交互才会恢复。

前提条件

- 请确保设备已经在中心管理服务（CMS）上注册，具体可参见 [设备注册](#)。
- 请确保 ISUPSDK 已升级至 V2.3.5.1 或更高版本。

操作步骤



说明

版本低于 V2.3.5.1 的 ISUPSDK 不支持休眠唤醒接口，接入休眠唤醒设备后存在兼容性问题。

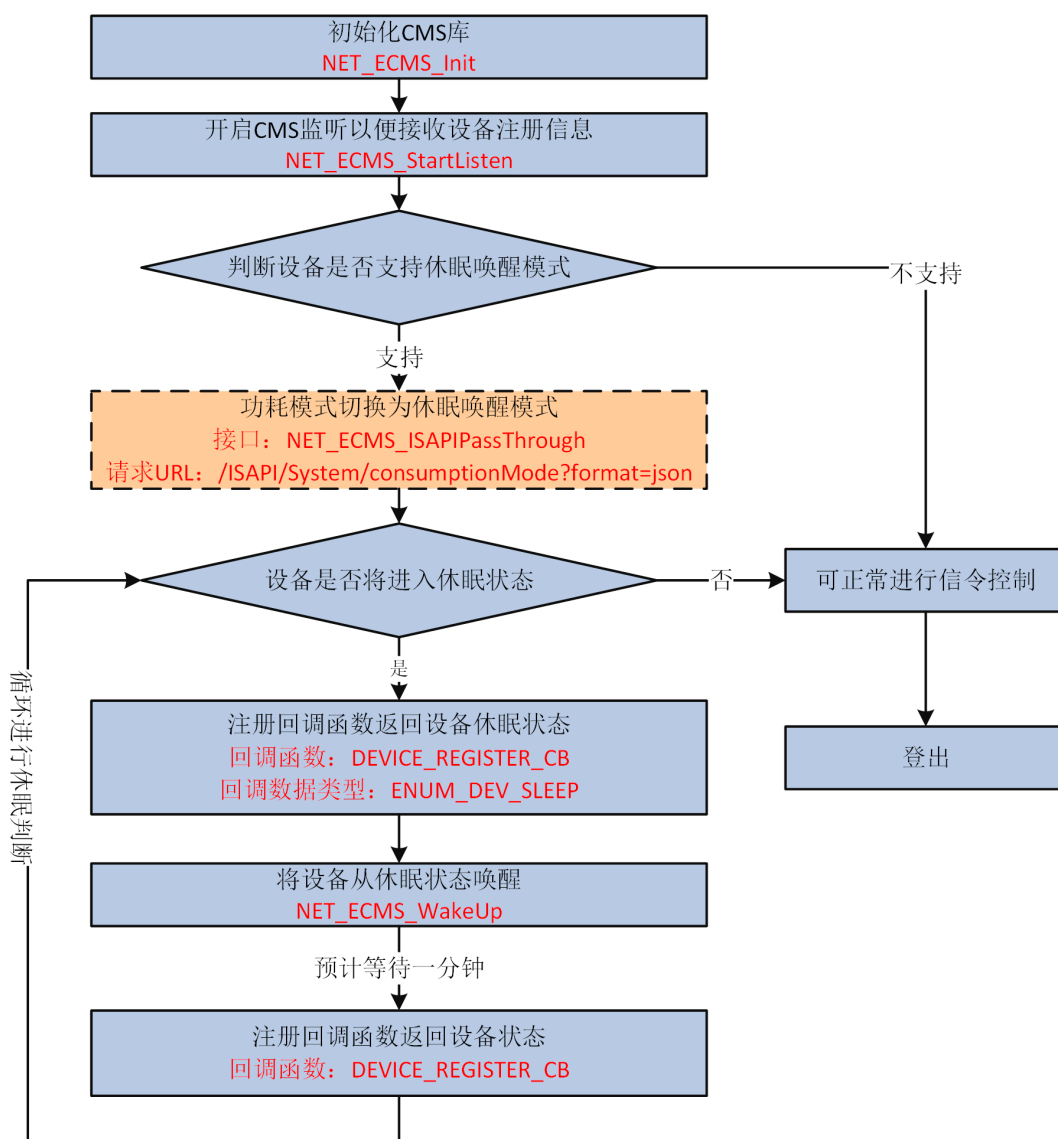


图 12-1 配置功耗模式的接口调用流程图

1. 调用 `NET_ECMS_ISAPIPassThrough` 传输请求 URI : `GET /ISAPI/System/capabilities` 获取设备能力集，来判断设备是否支持功耗模式配置。

设备能力集 `XML_DeviceCap` 通过结构体 `NET_EHOME_PTXML_PARAM` 由 `pOutBuffer` 返回。

若报文中返回节点 `<isSupportConsumptionMode>` 且其值为 `true`，表示设备支持功耗模式配置，可以继续操作后续步骤。

否则说明设备不支持功耗模式配置，请停止执行该任务。

2. 调用 `NET_ECMS_ISAPIPassThrough` 传输请求 URI : `GET /ISAPI/System/consumptionMode/capabilities?format=json` 获取功耗模式配置能力，来判断设备是否支持休眠唤醒模式。

功耗模式配置能力报文 *JSON Cap ConsumptionMode* 通过结构体 *NET_EHOME_PTXML_PARAM* 由 *pOutBuffer* 返回。

若报文中节点 *devWorkMode* 支持 “sleepOrWakeup”，表示设备支持休眠唤醒模式。

3. 可选操作: 调用 *NET_ECMS_ISAPIPassThrough* 传输请求 URI : *PUT /ISAPI/System/consumptionMode?format=json* , 并将请求报文 *JSON_ConsumptionMode* 中 *devWorkMode* 的值设为 “sleepOrWakeup”。



说明

设备默认的功耗模式为休眠唤醒模式。若休眠唤醒模式已开启，可忽略该步骤。

设备的功耗模式切换为休眠唤醒模式。

4. 设备休眠状态会通过回调数据类型 “ENUM_DEV_SLEEP” 回调至注册回调函数 *DEVICE_REGISTER_CB* 。



说明

- 若 5 分钟内未对设备执行任何操作，或者设备电量低于设定的阈值，设备将进入休眠状态。
- 设备处于休眠状态下不会响应任何信令控制（除休眠唤醒接口外），信令接口控制都将返回错误码 8301- “NET_ERR_DEV_SLEEP”。

5. 调用 *NET_ECMS_WakeUp* 将设备从休眠中唤醒。



说明

- 不同的设备，唤醒所需时长不同，一般预计为 1 分钟。
- 部分设备（如通过 Wi-Fi 连接的设备）心跳超时时长为 90 秒，若唤醒时长超过该时间，设备会自动下线，并回调设备下线状态，用户需重新注册设备。

设备将重新上电并注册，设备状态会回调至注册回调函数 *DEVICE_REGISTER_CB* 。

后续处理

- 调用 *NET_ECMS_StopListen* 停止 CMS 监听服务。
- 调用 *NET_ECMS_Fini* 释放被 CMS 占用的资源。

第 13 章 通过传输文本协议进行集成

ISUPSDK 支持通过传输文本协议进行集成。文本协议内容包括请求方法、请求 URI、查询参数、请求和响应报文。设备和平台（或系统）之间无需其他操作即可实现应用集成。

前提条件

请确保已调用 `NET_ECMS_Init` 初始化 CMS。

操作步骤

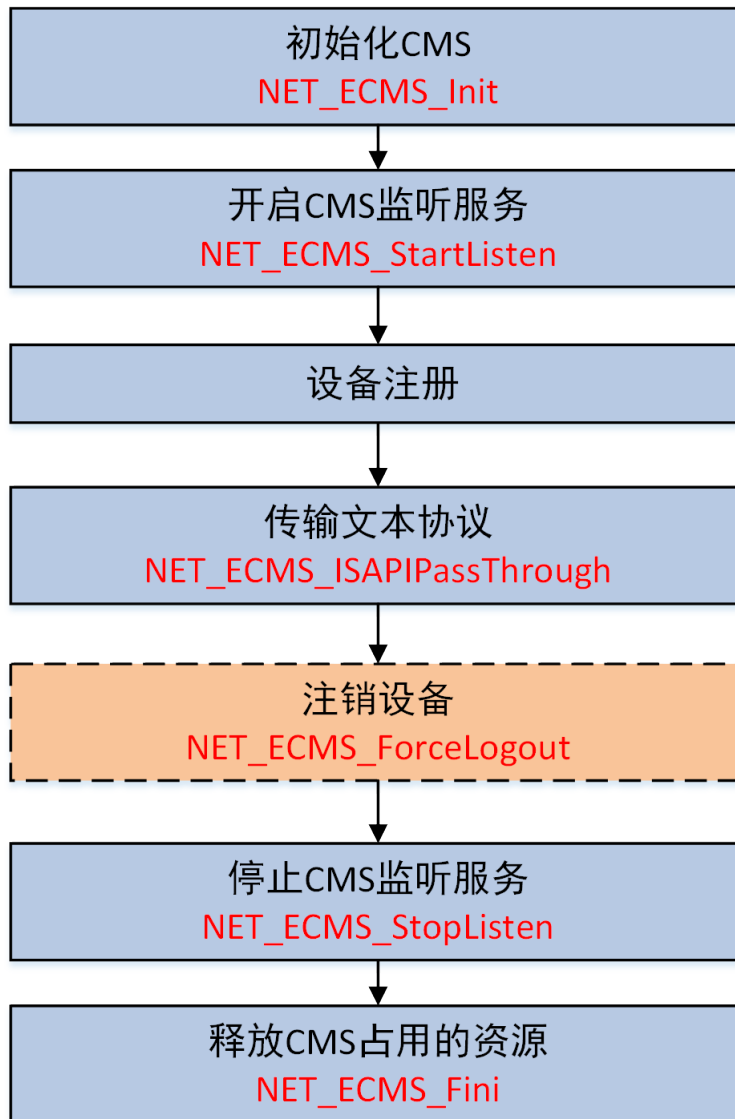


图 13-1 通过传输文本协议进行集成的接口调用流程

1. 调用 `NET_ECMS_StartListen` 开启 CMS 监听服务以便接收设备注册信息。



设备注册信息将在注册回调函数（`DEVICE_REGISTER_CB`）返回的结构体 `NET_EHOME_DEV_REG_INFO` 中查看。

2. 注册设备信息。



设备注册的消息处理流程，可参见 [设备注册消息处理流程](#)。

3. 调用 `NET_ECMS_ISAPIPassThrough` 传输文本协议，包括请求方法、请求 URI、查询参数、请求和响应报文，实现对应的应用。
4. 可选操作: 调用 `NET_ECMS_ForceLogout` 强制注销设备。

后续处理

- 调用 `NET_ECMS_StopListen` 停止 CMS 监听服务。
- 调用 `NET_ECMS_Fini` 释放被 CMS 占用的资源。

第 14 章 接口参考

14.1 注册模块

14.1.1 基础接口

NET_ECMS_Init

初始化注册模块库。

接口定义

```
BOOL CALLBACK NET_ECMS_Init(  
);
```

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

- 如果初始化失败，请检查加载库文件的路径和系统环境。
- 该接口必须与 *NET_ECMS_Fini* 配套调用。
- 该接口调用之前可以调用 *NET_ECMS_SetSDKInitCfg* 接口，其他接口均需要在该接口之后调用。

NET_ECMS_Fini

反初始化注册库并释放中央管理服务器(CMS)占用的资源。

接口定义

```
BOOL CALLBACK NET_ECMS_Fini(  
);
```

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

该接口必须与 *NET_ECMS_Init* 配套调用。

NET_ECMS_ForceLogout

强制注销设备且设备状态变为离线。

接口定义

```
BOOL CALLBACK NET_ECMS_ForceLogout(  
    LONG    lUserID  
);
```

参数

lUserID

[IN] 用户 ID，由 NET_ECMS_StartListen 配置的回调函数返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

备注

- 如果调用该接口强制注销了设备（设备状态变为离线），那么 CMS 将不再响应任何带有该设备用户 ID 的请求。一旦设备状态变为在线，注册请求将会被回调至用户，由用户决定设备是否已注册。
- 当 CMS 接收到设备离线的回调信息时，建议调用该接口释放资源。

NET_ECMS_SetSDKInitCfg

设置中心管理服务器（CMS）初始化参数。

接口定义

```
BOOL CALLBACK NET_ECMS_SetSDKInitCfg(  
    NET_EHOME_CMS_INIT_CFG_TYPE  enumType,  
    void* const                    lpInBuff  
);
```

参数

enumType

[IN] 初始化配置类型，详见枚举 NET_EHOME_CMS_INIT_CFG_TYPE。

lpInBuff

[IN] 初始化配置参数，取决于配置类型（enumType），参见对应关系如下。

enumType	lpInBuff
NET_EHOME_CMS_INIT_CFG_LIBEAY_PATH	带有 OpenSSL 加密库库名的路径。
NET_EHOME_CMS_INIT_CFG_SSLEAY_PATH	带有 OpenSSL 通信库库名的路径。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

- 该接口必须在初始化 *NET_ECMS_Init* 之前调用，否则 SDK 不能保证设置生效。
- 如果调用失败，无法获取错误码或获取正确的错误码，因为错误码必须在初始化之后才能获取。
- 该接口不会检验 OpenSSL 路径的有效性。加载库时会验证该路径，如果路径不正确，其他相关接口不会报错，错误会写入日志。
- 多个模块用于同一进程时，建议为所有模块设置相同的 OpenSSL 路径。例如，如果注册模块（CMS）以及报警管理模块（AMS）用于同一进程，如果 CMS 已加载 OpenSSL，但 AMS 的 OpenSSL 路径与 CMS 的 OpenSSL 路径不同，且库的版本号也不同，AMS 将使用已加载的 OpenSSL。
- 不同模块（AMS/CMS/SS/SMS）位于统一进程中，加载 OpenSSL 库时，应保证不同模块使用的 OpenSSL 库路径有效，且 OpenSSL 库版本和库名字相同。

NET_ECMS_GetBuildVersion

获取中央管理服务器（CMS）的版本信息。

接口定义

```
DWORD CALLBACK NET_ECMS_GetBuildVersion(
);
```

返回值

返回版本号，高两位字节表示主版本，低两位字节表示次版本，例如：0x02040001（2.4.0.1 版本）。

NET_ECMS_SetLogToFile

设置启用中心管理服务器（CMS）日志功能的参数。

接口定义

```
BOOL CALLBACK NET_ECMS_SetLogToFile (  
    DWORD      nLogLevel,  
    char       *strLogDir,  
    BOOL       bAutoDel  
);
```

参数

nLogLevel

[IN] 日志类型：0-启用日志功能（默认），1-错误日志，2-错误和调试日志，3-错误、调试和信息日志。

strLogDir

[IN] 日志文件保存路径。Windows 操作系统中的默认保存路径为“C:\\SdkLog\\”，Linux 操作系统中的默认保存路径为“/home/sdklog/”。如果要使用默认保存路径，请将该参数设置为“NULL”。

bAutoDel

[IN] 是否在一段时间后自动删除日志文件：“TRUE”-是（默认），“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩（将这生成的 50 个日志文件压缩成一个压缩文件）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

- 日志文件的保存路径必须是绝对路径，且必须以“\\”结尾，例如：“C:\\SdkLog\\”。
- 如果更改了保存路径，更改后的路径将用于保存下一个日志文件。
- 当 **bAutoDel** 的值为“TRUE”时，第一个日志文件将不会被删除，因为该文件包含起始时间、版本信息和其他重要信息。
- 允许 HCISUPSDK 在运行时，进行日志参数的修改。可在运行目录下添加 HCISUPSDK_Log_Switch.xml 文件来实现运行时日志参数的配置，可参见 [日志参数配置](#)。

NET_ECMS_GetSDKLocalCfg

获取中心管理服务器（CMS）的本地配置参数。

接口定义

```

BOOL CALLBACK NET_ECMS_GetSDKLocalCfg(
    NET_EHOME_LOCAL_CFG_TYPE    enumType,
    void*                        lpOutBuff
);

```

参数

enumType

[IN] 支持的本地配置类型宏定义值：-1, 0, 1, 2, 3, 4, 5 和 6，参见枚举定义 NET_EHOME_LOCAL_CFG_TYPE。

lpOutBuff

[OUT] 本地配置参数，与本地配置类型（enumType）有关，详情参见下表。

enumType	lpOutBuff	说明
UNDEFINE	未定义。	未定义。
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u> 。	获取 CMS 模块设备主动接入的安全性信息。
AMS_ADDRESS	<u>NET_EHOME_AMS_ADDRESS</u> 。	获取 CMS 报警的监听地址。
SEND_PARAM	<u>NET_EHOME_SEND_PARAM</u> 。	获取 CMS 模块的数据发送参数（如超时时间等）。
COM_PATH	COM 路径，指向一个字符串缓冲区。	获取 CMS 模块的 HCAapSDKCom（为 HCISUP 相关依赖库的目录）的路径。
SESSIONKEY_REQ_MOD	指向一个 BOOL 形变量。	获取是否进行 SessionKey 请求回调（只针对 ISUP5.0 协议设备）。true 表示在 ISUP5.0 设备注册回调中进行 ENUM_DEV_

enumType	lpOutBuff	说明
		SESSIONKEY_REQ 回调； false 表示在 ISUP5.0 设备注册回调中不产生 ENUM_DEV_SESSIONKEY_REQ 回调。 默认为 false。
DEV_DAS_PINGREQ_CALLBACK	<u>NET_EHOME_LOCAL_DEV_PINGREQ</u>	获取是否在注册回调中进行 ENUM_DEV_DAS_PINGREQ 回调，只针对 ISUP5.0 协议设备，默认不进行回调。
REGISTER_LISTEN_MODE	<u>NET_EHOME_REGISTER_LISTEN_MODE</u>	获取 CMS 模块的注册监听模式。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

参见

NET_ECMS_SetSDKLocalCfg

NET_ECMS_SetSDKLocalCfg

设置中心管理服务器（CMS）的本地配置参数。

接口定义

```

BOOL CALLBACK NET_ECMS_SetSDKLocalCfg(
    NET_EHOME_LOCAL_CFG_TYPE    enumType,
    void* const                  lpInBuff
);

```

参数

enumType

[IN] 支持的本地配置类型宏定义值：0, 1, 2, 3, 4, 5 和 6, 参见枚举定义 NET_EHOME_LOCAL_CFG_TYPE。

IpInBuff

[IN] 本地配置参数, 与本地配置类型 (**enumType**) 相关, 详见下表。

enumType	IpOutBuff	说明
UNDEFINE	未定义	未定义。
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u> 。	设置 CMS 模块设备主动接入的安全性信息。
AMS_ADDRESS	<u>NET_EHOME_AMS_ADDRESS</u> 。	设置 CMS 报警的监听地址。
SEND_PARAM	设置网络数据发送参数, 对应的结构为 <u>NET_EHOME_SEND_PARAM</u> 。	设置 CMS 模块的数据发送参数 (如超时时间等)。
SET_REREGISTER_MODE	设置 2.0 协议设备的重复注册模式, 对应的结构为 <u>NET_EHOME_SET_REREGISTER_MODE</u> 。	设置设备重复注册模式, 仅针对 ISUP2.0 协议设备。
COM_PATH	COM 路径, 指向一个字符串缓冲区 (例如: “./HCApSDKCom/”)。	设置 CMS 模块的 HCApSDKCom (为 HCISUP 相关依赖库的目录) 的路径。
SESSIONKEY_REQ_MOD	指向一个 BOOL 形变量。	设置是否进行 SessionKey 请求回调 (只针对 ISUP5.0 协议设备)。true 表示在 ISUP5.0 设备注册回调中进行 ENUM_DEV_SESSIONKEY_REQ 回调; false 表示在 ISUP5.0 设

enumType	IpOutBuff	说明
		备注册回调中不进行 ENUM_DEV_SESSIONKEY_REQ 回调。默认为 false。
DEV_DAS_PINGREO_CALLBACK	<u>NET_EHOME_LOCAL_DEV_PINGREO</u>	设置是否在注册回调中进行 ENUM_DEV_DAS_PINGREO 回调，只针对 ISUP5.0 协议设备，默认不进行回调。
REGISTER_LISTEN_MODE	<u>NET_EHOME_REGISTER_LISTEN_MODE</u>	设置 CMS 模块的注册监听模式。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

参见

NET_ECMS_GetSDKLocalCfg

NET_ECMS_GetLastError

如果调用失败或完成，获取错误码。

接口定义

```
DWORD CALLBACK NET_ECMS_GetLastError(
);
```

返回值

返回错误码，详见 ISUPSDK 错误码。

备注

该接口需要在 NET_ECMS_Init 之后调用。

NET_ECMS_WakeUp

将设备从休眠状态唤醒

接口定义

```
BOOL CALLBACK NET_ECMS_WakeUp(  
    LONG          iUserID  
);
```

参数

iUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

14.1.2 配置和控制接口

NET_ECMS_GetDeviceSessionKey

获取支持 5.0 版本 ISUP 的设备会话密钥信息。

接口定义

```
BOOL CALLBACK NET_ECMS_GetDeviceSessionKey(  
    NET_EHOME_DEV_SESSIONKEY *deviceKey  
);
```

参数

deviceKey

[OUT] 支持 5.0 版本 ISUP 的设备会话密钥信息，详情参见结构体

[NET_EHOME_DEV_SESSIONKEY](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

参见

[NET_ECMS_SetDeviceSessionKey](#)

NET_ECMS_SetDeviceSessionKey

设置支持 5.0 版本 ISUP 的设备会话密钥。

接口定义

```
BOOL CALLBACK NET_ECMS_SetDeviceSessionKey(  
    NET_EHOME_DEV_SESSIONKEY    *deviceKey  
);
```

参数

deviceKey

[IN] 支持 5.0 版本 ISUP 的设备会话密钥信息，详情参见结构体 [NET_EHOME_DEV_SESSIONKEY](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

- 一般在注册回调的 `ENUM_DEV_SESSIONKEY` 时进行调用，以便 5.0 协议设备完成注册。
- 没有负载均衡（LBS）时，一般在 CMS 注册回调 `ENUM_DEV_SESSIONKEY` 时，调用该接口。
- 带负载均衡（LBS）时，一般在设备接入服务（DAS）注册回调 `ENUM_DEV_SESSIONKEY_REQ` 时，调用该接口（此时 LBS 将 `SeesionKey` 通知给 DAS）。

参见

[NET_ECMS_GetDeviceSessionKey](#)

NET_ECMS_GetDevConfig

获取设备参数。

接口定义

```
BOOL CALLBACK NET_ECMS_GetDevConfig(  
    LONG                lUserID,  
    DWORD               dwCommand,  
    NET_EHOME_CONFIG    *lpConfig,
```

```
DWORD
dwConfigSize
);
```

参数

iUserID

[IN] 用户 ID, 由 NET_ECMS_StartListen 注册的注册回调函数返回。

dwCommand

[IN] 配置命令。

IpConfig

{IN&OUT} 输入或输出配置参数, 不同的配置命令对应于不同的输入或输出配置参数。请参见结构体 NET_EHOME_CONFIG 了解参数格式。

dwConfigSize

[IN] IpConfig 指向的结构体的大小。

返回值

返回 *TRUE* 成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 NET_ECMS_GetLastError 获取错误码。

备注

不同的命令对应于不同的条件和输出参数。

dwCommand	IpConfig → pCondBuf	IpConfig → pOutBuf	命令号
获取设备信息的命令: NET_EHOME_GET_DEVICE_INFO	NULL	<u>NET_EHOME_DEVICE_INFO</u>	1
获取版本信息的命令: NET_EHOME_GET_VERSION_INFO	NULL	<u>NET_EHOME_VERSION_INFO</u>	2
获取设备基本信息的命令: NET_EHOME_GET_DEVICE_CFG	NULL	<u>NET_EHOME_DEVICE_CFG</u>	3
获取网络参数的命令: NET_EHOME_GET_NETWORK_CFG	NULL	<u>NET_EHOME_NETWORK_CFG</u>	5

dwCommand	<i>lpConfigp</i> → <i>pCondBuf</i>	<i>lpConfigp</i> → <i>pOutBuf</i>	命令号
获取编码参数的命令: NET_EHOME_GET_COMPRESSION_CFG	<u>NET_EHOME_COMPRESSION_COND</u>	<u>NET_EHOME_COMPRESSION_CFG</u>	7
获取图像参数的命令: NET_EHOME_GET_IMAGE_CFG	4 字节通道号	<u>NET_EHOME_IMAGE_CFG</u>	9
获取报警输入参数的命令: NET_EHOME_GET_ALARMIN_CFG	<u>NET_EHOME_ALARMIN_COND</u>	<u>NET_EHOME_ALARMIN_CFG</u>	11
获取布防计划参数的命令: NET_EHOME_GET_ALARM_TIME_CFG	<u>NET_EHOME_ALARM_TIME_COND</u>	<u>NET_EHOME_ALARM_TIME_CFG</u>	13
获取报警输出参数的命令: NET_EHOME_GET_ALARMOUT_CFG	4 字节报警输出号, 从 1 开始。	<u>NET_EHOME_ALARMOUT_CFG</u>	15
获取报警输出状态的命令: NET_EHOME_GET_ALARMOUT_STATUS_CFG	4 字节报警输出号, 从 1 开始。	<u>NET_EHOME_ALARMOUT_STATUS_CFG</u>	17
获取 OSD (屏幕菜单式调节方式) 参数的命令: NET_EHOME_GET_PIC_CFG	4 字节通道号	<u>NET_EHOME_PIC_CFG</u>	22

参见

NET_ECMS_SetDevConfig

NET_ECMS_SetDevConfig

设置设备参数。

接口定义

```

BOOL CALLBACK NET_ECMS_SetDevConfig(
    LONG                iUserID,
    DWORD               dwCommand,
    NET_EHOME_CONFIG   *lpConfig,
    DWORD               dwConfigSize
);

```

参数

iUserID

[IN] 用户 ID, 由 [NET_ECMS_StartListen](#) 注册的注册回调函数返回。

dwCommand

[IN] 配置命令。

lpConfig

{IN&OUT} 输入或输出配置参数, 不同的配置命令对应于不同的输入或输出配置参数。请参见结构体 [NET_EHOME_CONFIG](#) 了解参数格式。

dwConfigSize

[IN] [lpConfig](#) 指向的结构体的大小。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

不同的命令对应于不同的条件和输出参数。

dwCommand	lpConfigp → pCondBuf	lpConfigp → plnBuf	命令号
设置设备信息的命令: NET_EHOME_SET_DEVICE_CFG	NULL	NET_EHOME_DEVICE_CFG	4
设置网络参数的命令: NET_EHOME_SET_NETWORK_CFG	NULL	NET_EHOME_NETWORK_CFG	6

dwCommand	<i>IpConfigp</i> → <i>pCondBuf</i>	<i>IpConfigp</i> → <i>pInBuf</i>	命令号
设置编码参数的命令: NET_EHOME_SET_COMPRESSION_CFG	<u>NET_EHOME_COMPRESSION_COND</u>	<u>NET_EHOME_COMPRESS ION_CFG</u>	8
设置图像参数的命令: NET_EHOME_SET_IMAGE_CFG	4 字节通道号	<u>NET_EHOME_IMAGE_CFG</u>	10
设置报警输入参数的命令: NET_EHOME_SET_ALARMIN_CFG	<u>NET_EHOME_ALARMIN_COND</u>	<u>NET_EHOME_ALARMIN_CFG</u>	12
设置布防计划参数的命令: NET_EHOME_SET_ALARM_TIME_CFG	<u>NET_EHOME_ALARM_TIME_COND</u>	<u>NET_EHOME_ALARM_TIME_CFG</u>	14
设置报警输出参数的命令: NET_EHOME_SET_ALARMOUT_CFG	4 字节报警输出号, 从 1 开始。	<u>NET_EHOME_ALARMOUT_CFG</u>	16
设置报警输出状态的命令: NET_EHOME_SET_ALARMOUT_STATUS_CFG	4 字节报警输出号, 从 1 开始。	<u>NET_EHOME_ALARMOUT_STATUS_CFG</u>	18
设置 OSD (屏幕菜单式调节方式) 参数的命令: NET_EHOME_GET_PIC_CFG	4 字节通道号	<u>NET_EHOME_PIC_CFG</u>	23

参见

NET_ECMS_GetDevConfig

NET_ECMS_ResetDevRegisterState

重置支持 5.0 版本 ISUP 的设备注册状态。

接口定义

```
BOOL CALLBACK NET_ECMS_ResetDevRegisterState(
    const char    *sDeviceID
)
```

参数

sDeviceID

[IN] 设备 ID。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_RemoteControl

执行远程控制。

接口定义

```
BOOL CALLBACK NET_ECMS_RemoteControl(
    LONG                lUserID,
    DWORD               dwCommand,
    NET_EHOME_REMOTE_CTRL_PARAM *lpCtrlParam
);
```

参数

lUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

dwCommand

[IN] 控制命令，详情参见下表。

dwCommand	描述	命令号
NET_EHOME_MANUAL_IOOUT	手动控制报警输出。	19
NET_EHOME_PTZ_CTRL	控制 PTZ。	1000
NET_EHOME_PRESET_CTRL	设置、删除或调用预置点。	1001

dwCommand	描述	命令号
NET_EHOME_PZIN	放大或缩小。	1002
NET_EHOME_PTRACK	3D 定位。	1003

IpCtrlParam

[IN] 控制参数，取决于控制命令，详情参见结构体 [NET_EHOME_REMOTE_CTRL_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_SetCallback

为中心管理服务器（CMS）注册异步回调函数。

接口定义

```

BOOL CALLBACK NET_ECMS_SetCallback(
    NET_EHOME_CALLBACK_TYPE    enumCallbackType,
    ECMSCallback                fnCallback,
    void*                        pUser
);

```

参数**enumCallbackType**

[IN] 回调类型，详见枚举 [NET_EHOME_CALLBACK_TYPE](#)。

fnCallback

[IN] 回调函数，详情参见 [ECMSCallback](#)。

pUser

[IN] 用户参数指针。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_SetExceptionCallBack

注册中心管理服务器（CMS）的异常回调函数。

接口定义

```
BOOL CALLBACK NET_ECMS_SetExceptionCallBack(  
    DWORD                dwMessage,  
    HANDLE               hWnd,  
    fExceptionCallBack  cbExceptionCallBack)  
void *pUser  
    DWORD                dwType,  
    LONG                 iUserID,  
    LONG                 iHandle,  
    void *pUser  
)
```

参数

dwMessage

[IN] 消息类型（该参数仅对 Windows 操作系统有效）。

hWnd

[IN] 窗口句柄（该参数仅对 Windows 操作系统有效）。

cbExceptionCallBack

[IN] 异常回调函数，详情参见 [fExceptionCallBack](#)。

pUser

[IN] 用户指针。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_LongConfigCreate

建立长连接。

接口定义

```
LONG CALLBACK NET_ECMS_LongConfigCreate(  
    LONG                lUserID,  
    NET_EHOME_LONG_CFG_INPUT *pLongCfgInput  
)
```

参数

IUserID

[IN] 用户 ID。

pLongCfgInput

[IN] 长连接参数，详情参见结构体 [NET_EHOME_LONG_CFG_INPUT](#)。

返回值

返回大于等于 0 的数值表示成功，返回小于 0 的数值表示失败。
如果返回值小于 0，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_LongConfigDestory

销毁长连接。

接口定义

```
BOOL CALLBACK NET_ECMS_LongConfigDestory(  
    LONG lHandle  
)
```

参数

lHandle

长连接句柄，由 [NET_ECMS_LongConfigCreate](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。
如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_LongConfigGetSessionId

根据长连接句柄获取会话 ID。

接口定义

```
BOOL CALLBACK NET_ECMS_LongConfigGetSessionId(  
    LONG iHandle,  
    LONG *pSessionId  
)
```

参数

iHandle

长连接句柄，由 [NET_ECMS LongConfigCreate](#) 返回。

pSessionId

会话 ID 指针。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS GetLastError](#) 获取错误码。

NET_ECMS_LongConfigSend

通过长连接发送数据。

接口定义

```
BOOL CALLBACK NET_ECMS_LongConfigSend(  
    LONG                                lHandle,  
    NET_EHOME_LONG_CFG_SEND            *pSend  
)
```

参数

lHandle

长连接句柄，由 [NET_ECMS LongConfigCreate](#) 返回。

pSend

[IN] 要通过长连接发送的参数，详情参见结构体 [NET_EHOME_LONG_CFG_SEND](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS GetLastError](#) 获取错误码。

NET_ECMS_MakeIFrame

强制 I 帧。

接口定义

```
BOOL CALLBACK NET_ECMS_MakeIFrame(  
    LONG                                iUserID,
```

```
NET_EHOME_MAKE_I_FRAME *pIFrameParma  
);
```

参数

iUserID

[IN] 用户 ID，由 NET_ECMS StartListen 注册的回调函数返回。

pIFrameParma

[IN] 强制 I 帧的参数，详情参见结构体 NET_EHOME_MAKE_I_FRAME。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS GetLastError 获取错误码。

备注

该接口只支持 ISUP5.0 协议设备。

NET_ECMS_STDBinaryCtrl

控制二进制参数。

接口定义

```
BOOL CALLBACK NET_ECMS_STDBinaryCtrl(  
    LONG                lUserID,  
    LONG                dwCommand,  
    NET_EHOME_BINARY_SEND_DATA *pSendData  
);
```

参数

lUserID

[IN] 用户 ID。

dwCommand

[IN] 控制命令，详情参见下表。

命令	命令号	描述
NET_EHOME_S2C_CANCEL_SCHEDULE	0x65	取消发布计划。
NET_EHOME_S2C_SEND_ERRCODE	0xD	发送错误码。
NET_EHOME_S2C_SET_TERMINAL_NAME	0x71	设置终端名。
NET_EHOME_S2C_LOGOUT	0x20	通知终端登出服务器。
NET_EHOME_S2C_RETRANSMIT_TO_TERM	0x108	将数据从服务器传输至终端。

pSendData

[IN] 用于发送数据的参数，详情参见结构体 [NET_EHOME_BINARY_SEND_DATA](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_STDBinaryCfg

配置二进制参数。

接口定义

```

BOOL CALLBACK NET_ECMS_STDBinaryCfg (
    LONG                lUserID,
    LONG                dwCommand,
    NET_EHOME_BINARY_SEND_DATA *pSendData,
    NET_EHOME_BINARY_RECV_DATA *pRecvData
)

```

参数**lUserID**

[IN] 用户 ID。

dwCommand

[IN] 配置命令，详情参见下表。

命令	命令号	描述
NET_EHOME_S2C_TIME_ADJUST	0x9	同步终端时间。
NET_EHOME_S2C_SET_SERVER_INFO	0x18	设置服务器信息。
NET_EHOME_S2C_GET_TERMINAL_INFO	0x19	获取终端信息。
NET_EHOME_S2C_POST_SCHEDULE	0x50	下发计划。
NET_EHOME_S2C_TERMINALCONTROL_V20	0x51	控制终端（V20）。
NET_EHOME_S2C_TERMINALCONTROL	0x53	控制终端。
NET_EHOME_S2C_PLAYCONTROL	0x54	控制终端播放。
NET_EHOME_S2C_SET_IPCINFO	0x55	设置网络摄像机信息。
NET_EHOME_S2C_GET_WORKSTATUS	0x56	获取设备工作状态。
NET_EHOME_S2C_VERSIONUPGRADE	0x57	升级设备。
NET_EHOME_S2C_SETPLAYERPARAM	0x58	设置终端的播放参数。
NET_EHOME_S2C_SETSERVERADDR	0x5B	设置终端注册的服务器信息。
NET_EHOME_WEATHERINFO	0x61	天气信息。
NET_EHOME_S2C_GET_SCREENSHOT	0x62	获取截取的终端屏幕图片。
NET_EHOME_S2C_GET_DEFAULTPARAM	0x63	获取默认设置。

命令	命令号	描述
NET_EHOME_S2C_REPLACE_MATERIAL	0x64	更改素材。
NET_EHOME_S2C_SET_QUEUE_INFO	0x66	设置呼叫管理信息。
NET_EHOME_S2C_PUBLISH_SHOT_PIC	0x67	下发弹图。
NET_EHOME_S2C_GET_IPCINFO	0x68	获取网络摄像机信息。
NET_EHOME_S2C_SET_TERMINAL_IP	0x69	设置终端 IP 地址。
NET_EHOME_S2C_GET_TERMINAL_IP	0x70	获取终端 IP 地址。
NET_EHOME_S2C_SET_SWITCHPLAN	0x100	下发开/关机计划。
NET_EHOME_S2C_GET_SWITCHPLAN	0x101	获取开/关机计划。
NET_EHOME_S2C_SET_VOLUMEPLAN	0x102	设置音量调节计划。
NET_EHOME_S2C_GET_VOLUMEPLAN	0x103	获取音量调节计划。
NET_EHOME_S2C_SET_CHARACTER	0x106	向终端发送文本消息。
NET_EHOME_S2C_GET_CHARACTER	0x107	从终端获取文本消息。
NET_EHOME_S2C_SET_ADB_CFG	0x110	设置 ADB 参数。
NET_EHOME_S2C_GET_ADB_CFG	0x111	获取 ADB 参数。
NET_EHOME_S2C_SET_TIEM_ZONE	0x112	设置终端时区。
NET_EHOME_S2C_GET_TIME_ZONE	0x113	获取终端时区。
NET_EHOME_S2C_SET_SADP	0x114	设置终端 SADP 信息。

命令	命令号	描述
NET_EHOME_S2C_GET_SADP	0x115	获取终端 SADP 信息。
NET_EHOME_S2C_GET_COMPONENT	0x116	获取终端组件信息。
NET_EHOME_S2C_TRANS_WITH_RET	0x10A	在服务器和终端之间传输数据。
NET_EHOME_S2C_PUBLISH_XML	0x200	传输计划、插播、升级的 XML 数据。
NET_EHOME_S2C_SCREEN_SHOT	0x201	截取终端屏幕。

pSendData

[IN] 用于发送数据的参数，详情参见结构体 NET_EHOME_BINARY_SEND_DATA。

pRecvData

[IN] 用于接收数据的参数，详情参见结构体 NET_EHOME_BINARY_RECV_DATA。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

NET_ECMS_SetAliveTimeout

设置 ISUPSDK 对某个 5.0 协议设备的心跳超时时间（不会影响设备的心跳发送，只影响 ISUPSDK 的心跳检测）。

接口定义

```

BOOL CALLBACK NET_ECMS_SetAliveTimeout (
    LONG    lUserID,
    DWORD   dwKeepAliveSec,
    DWORD   dwTimeOutCount
)

```

参数**lUserID**

[IN] 用户 ID，由 NET_ECMS_StartListen 注册的回调函数返回。

dwKeepAliveSec

单次心跳超时时间，单位为秒。

dwTimeOutCount

心跳超时次数。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。若返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

实际的心跳超时时间为 **dwKeepAliveSec** 乘以 **dwTimeOutCount** 的值。

14.1.3 监听接口

NET_ECMS_StartListen

启用中心管理服务器（CMS）的监听并注册回调函数以接收设备注册信息。

接口定义

```
LONG CALLBACK NET_ECMS_StartListen(  
    NET_EHOME_CMS_LISTEN_PARAM    *lpCMSListenPara  
);
```

参数

lpCMSListenPara

[IN] 监听参数，详情参见结构体 [NET_EHOME_CMS_LISTEN_PARAM](#)。

返回值

返回-1 表示失败，返回其他值表示 [NET_ECMS_StopListen](#) 的句柄参数。
如果返回-1，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StopListen

停止中心管理服务器（CMS）的监听。

接口定义

```
BOOL CALLBACK NET_ECMS_StopListen(  
    LONG    lHandle  
);
```

参数

lHandle

[IN] 监听句柄，由 [NET_ECMS_StartListen](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StartListenProxy

开始监听代理类型。

接口定义

```
LONG CALLBACK NET_ECMS_StartListenProxy(  
    NET_EHOME_PT_PARAM *lpStru  
)
```

参数

lpStru

[IN] 监听代理类型的参数，详情参见结构体 [NET_EHOME_PT_PARAM](#)。

返回值

返回小于 0 的值表示失败，返回其他值表示 [NET_ECMS_StopListenProxy](#) 的监听句柄。

如果返回小于 0 的值，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StopListenProxy

停止监听代理类型。

接口定义

```
BOOL CALLBACK NET_ECMS_StopListenProxy(  
    LONG    lListenHandle,
```

```
DWORD dwProxyType
)
```

参数

IListenHandle

[IN] 监听句柄，由 [NET_ECMS_StartListenProxy](#) 返回。

dwProxyType

[IN] 代理类型，枚举于 [NET_CMS_ENUM_PROXY_TYPE](#)。默认的代理类型为海康私有协议代理（ENUM_PROXY_TYPE_NETSDK）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

参见

[NET_ECMS_StartListenProxy](#)

14.1.4 预览接口

NET_ECMS_StartGetRealStream

开始预览。

接口定义

```
BOOL CALLBACK NET_ECMS_StartGetRealStream(
    LONG lUserID,
    LPNET_EHOME_PREVIEWINFO_IN pPreviewInfoIn,
    LPNET_EHOME_PREVIEWINFO_OUT pPreviewInfoOut
);
```

参数

IUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pPreviewInfoIn

[IN] 预览请求的输入参数，详情参见结构体 [NET_EHOME_PREVIEWINFO_IN](#)。

pPreviewInfoOut

[OUT] 预览请求的输出参数，详情参见结构体 [NET_EHOME_PREVIEWINFO_OUT](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StartGetRealStreamV11

请求开始预览。

接口定义

```
BOOL CALLBACK NET_ECMS_StartGetRealStreamV11(  
    LONG                                lUserID,  
    NET_EHOME_PREVIEWWINFO_IN_V11     *pPreviewInfoIn,  
    NET_EHOME_PREVIEWWINFO_OUT       *pPreviewInfoOut  
);
```

参数

lUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pPreviewInfoIn

[IN] 预览请求的输入参数，详情参见结构体 [NET_EHOME_PREVIEWWINFO_IN_V11](#)。

pPreviewInfoOut

[OUT] 预览请求的输出参数，详情参见结构体 [NET_EHOME_PREVIEWWINFO_OUT](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

参见

[NET_ECMS_StartPushRealStream](#)

[NET_ECMS_StopGetRealStream](#)

NET_ECMS_StartPushRealStream

中心管理服务器（CMS）向设备发送请求，设备开始传输预览实时码流。

接口定义

```
BOOL CALLBACK NET_ECMS_StartPushRealStream(  
    LONG                                lUserID,
```

```
NET_EHOME_PUSHSTREAM_IN      *pPushInfoIn,  
NET_EHOME_PUSHSTREAM_OUT    *pPushInfoOut  
);
```

参数

IUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pPushInfoIn

[IN] 实时码流传输请求的输入参数，详情参见结构体 [NET_EHOME_PUSHSTREAM_IN](#)。

pPushInfoOut

[OUT] 实时码流传输请求的输出参数，详情参见结构体 [NET_EHOME_PUSHSTREAM_OUT](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

对 4.0 (不含) 以下 ISUP 协议版本设备取流不需要调用该接口。对 4.0 (含) 以上 ISUP 协议版本设备取流需要调用该接口。调用该接口前，确保 [NET_ECMS_StartGetRealStreamV11](#) 或 [NET_ECMS_StartGetRealStream](#) 已调用成功，确保 CMS 已接收预览请求响应中的设备会话 ID。

参见

[NET_ECMS_StartGetRealStreamV11](#)

NET_ECMS_StopGetRealStream

请求停止预览。

接口定义

```
BOOL CALLBACK NET_ECMS_StopGetRealStream(  
    int      iUserID,  
    int      iSessionID  
);
```

参数

iUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

iSessionID

[IN] 会话 ID，由 [NET_ECMS_StartGetRealStreamV11](#) 或 [NET_ECMS_StartGetRealStream](#) 返回。
由返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。
如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StopGetRealStreamEx

请求停止预览。该接口支持异步模式。

接口定义

```
BOOL CALLBACK NET_ECMS_StopGetRealStreamEx (  
    LONG                                lUserID,  
    NET_EHOME_STOPSTREAM_PARAM         *pStopParam  
)
```

参数

lUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pStopParam

[IN] 停止预览的请求参数，详情参见结构体 [NET_EHOME_STOPSTREAM_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。
如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

扩展自

[NET_ECMS_StopGetRealStream](#)

14.1.5 回放接口

NET_ECMS_StartFindFile

开始查找文件。

接口定义

```
LONG CALLBACK NET_ECMS_StartFindFile(  
    LONG                lUserID,  
    LPNET_EHOME_FINDCOND pFindCond  
)
```

参数

IUserID

[IN] 用户 ID, 由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pFindCond

[IN] 查找参数, 参见 [NET_EHOME_FINDCOND](#) 定义。

返回值

返回-1 表示失败, 返回其他值表示 [NET_ECMS_FindNextFile_V11](#) 和 [NET_ECMS_StopFindFile](#) 的句柄。

如果返回-1, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

该接口已被 [NET_ECMS_StartFindFile_V11](#) 替代, 建议使用高版本接口。

NET_ECMS_StartFindFile_V11

开始查找文件。

接口定义

```
LONG CALLBACK NET_ECMS_StartFindFile_V11(  
    LONG                lUserID,  
    LONG                lSearchType,  
    VOID                *pFindCond,  
    DWORD               dwCondSize  
) ;
```

参数

IUserID

[IN] 用户 ID, 由 [NET_ECMS_StartListen](#) 注册的注册回调函数返回。

ISearchType

[IN] 检索类型, 参见枚举定义 [SEARCH_TYPE](#)。

pFindCond

[IN] 保存检索条件的缓冲区，不同的检索类型对应于不同的检索条件，详情参见备注。

dwCondSize

[IN] 检索条件缓冲区大小。

返回值

返回-1表示失败，返回其他值表示 *NET_ECMS_FindNextFile_V11* 和 *NET_ECMS_StopFindFile* 的句柄。

如果返回-1，请调用 *NET_ECMS_GetLastError* 获取错误码。

备注

检索类型（**ISearchType**）和检索条件（**pFindCond**）之间的关系如下表所示。

ISearchType	pFindCond
ENUM_SEARCH_RECORD_FILE	<i>NET_EHOME_REC_FILE_COND</i>
ENUM_SEARCH_PICTURE_FILE	<i>NET_EHOME_PIC_FILE_COND</i>
ENUM_SEARCH_FLOW_INFO	<i>NET_EHOME_FLOW_COND</i>
ENUM_SEARCH_DEV_LOG	<i>NET_EHOME_DEV_LOG_COND</i>
ENUM_SEARCH_ALARM_HOST_LOG	<i>NET_EHOME_ALARM_HOST_LOG_COND</i>

NET_ECMS_FindNextFile

逐个获取检索到的文件（仅支持录像文件查询）。

接口定义

```
LONG CALLBACK NET_ECMS_FindNextFile(
    LONG                lHandle,
    LPNET_EHOME_FINDDATA pFindData
);
```

参数

lHandle

[IN] 检索句柄，由 *NET_ECMS_StartFindFile* 返回。

pFindData

[OUT] 返回的数据结构，见 *NET_EHOME_FINDDATA* 定义。

返回值

返回-1表示失败，返回其他值表示获取状态，详情参见下表。
如果返回-1，请调用 *NET_ECMS_GetLastError* 获取错误码。

状态名	状态码	描述
ENUM_GET_NEXT_STATUS_SUCCESS	1000	获取到一个文件。处理完该数据后，再次调用该接口获取下一个文件。
ENUM_GET_NETX_STATUS_NO_FILE	1001	未找到文件。
ENUM_GET_NETX_STATUS_NEED_WAIT	1002	正在检索中。请稍后。
ENUM_GET_NEXT_STATUS_FINISH	1003	没有更多文件。结束检索。
ENUM_GET_NEXT_STATUS_FAILED	1004	检索异常。
ENUM_GET_NEXT_STATUS_NOT_SUPPORT	1005	设备不支持该操作。

备注

该接口已被 *NET_ECMS_FindNextFile_V11* 替代，建议使用高版本接口

NET_ECMS_FindNextFile_V11

逐个获取检索到的文件。

接口定义

```
LONG CALLBACK NET_ECMS_FindNextFile_V11(
    LONG        lHandle,
    VOID        *pFindData,
    DWORD       dwDataSize
);
```

参数

lHandle

[IN] 检索句柄，由 *NET_ECMS_StartFindFile_V11* 返回。

pFindData

[OUT] 返回的数据结构，*NET_ECMS_StartFindFile_V11* 中配置的不同检索类型（**SearchType**）对应于不同的数据结构，详情参见下表。

SearchType	pFindData
检索视频文件：ENUM_SEARCH_RECORD_FILE	<u>NET_EHOME_REC_FILE</u>
检索图片文件：ENUM_SEARCH_PICTURE_FILE	<u>NET_EHOME_PIC_FILE</u>
检索流信息：ENUM_SEARCH_FLOW_INFO	<u>NET_EHOME_FLOW_INFO</u>
检索设备日志：ENUM_SEARCH_DEV_LOG	<u>NET_EHOME_DEV_LOG</u>
检索报警主机日志：ENUM_SEARCH_ALARM_HOST_LOG	<u>NET_EHOME_ALARM_HOST_LOG</u>

dwDataSize

[OUT] 返回数据的大小。

返回值

- 返回-1 表示失败，返回其他值表示获取状态，详情参见下表。

状态名	状态码	描述
ENUM_GET_NEXT_STATUS_SUCCESS	1000	获取到一个文件。处理完该数据后,再次调用该接口获取下一个文件。
ENUM_GET_NETX_STATUS_NO_FILE	1001	未找到文件。
ENUM_GET_NETX_STATUS_NEED_WAIT	1002	正在检索中。请稍后。
ENUM_GET_NEXT_STATUS_FINISH	1003	没有更多文件。结束检索。
ENUM_GET_NEXT_STATUS_FAILED	1004	检索异常。
ENUM_GET_NEXT_STATUS_NOT_SUPPORT	1005	设备不支持该操作。

- 如果返回-1, 请调用 NET_ECMS_GetLastError 获取错误码。

备注

- 该接口必须在调用 [NET_ECMS_StartFindFile_V11](#) 设置检索条件并获取检索句柄后才能调用。
- 若要获取所有检索结果,必须反复调用该接口。

NET_ECMS_StopFindFile

停止检索文件并释放资源。

接口定义

```
BOOL CALLBACK NET_ECMS_StopFindFile(  
    LONG    iHandle  
);
```

参数

iHandle

[IN] 检索句柄, 由 [NET_ECMS_StartFindFile_V11](#) 返回。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StartPlayBack

请求开始回放。

接口定义

```
BOOL CALLBACK NET_ECMS_StartPlayBack(  
    LONG                iUserID,  
    NET_EHOME_PLAYBACK_INFO_IN *pPlayBackInfoIn  
    NET_EHOME_PLAYBACK_INFO_OUT *PlayBackInfoOut  
);
```

参数

iUserID

[IN] 用户 ID, 由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pPlayBackInfoIn

[IN] 回放请求的输入参数, 详情参见结构体 [NET_EHOME_PLAYBACK_INFO_IN](#)。

pPlayBackInfoOut

[OUT] 回放请求的输出参数，详情参见结构体 [NET_EHOME_PLAYBACK_INFO_OUT](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

参见

[NET_ECMS_StartPushPlayBack](#)

[NET_ECMS_StopPlayBack](#)

NET_ECMS_StartPushPlayBack

中心管理服务器（CMS）向设备发送请求，设备开始传输回放码流。

接口定义

```
BOOL CALLBACK NET_ECMS_StartPushPlayBack(  
    LONG                                lUserID,  
    NET_EHOME_PUSHPLAYBACK_IN          *pPushInfoIn,  
    NET_EHOME_PUSHPLAYBACK_OUT         *pPushInfoOut  
);
```

参数

lUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pPushInfoIn

[IN] 回放码流传输请求的输入参数，详情参见结构体 [NET_EHOME_PUSHPLAYBACK_IN](#)。

pPushInfoOut

[OUT] 回放码流传输请求的输出参数，详情参见结构体 [NET_EHOME_PUSHPLAYBACK_OUT](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

对于支持 4.0 版本及以上 ISUP 的设备，在调用该接口之前，请确保 CMS 已接收回放请求响应中的设备会话 ID。

参见

[NET_ECMS_StartPlayBack](#)

NET_ECMS_PlayBackOperate

控制回放，例如暂停回放和恢复回放。

接口定义

```
BOOL CALLBACK NET_ECMS_PlayBackOperate (
    LONG                iUserID,
    ENUM_NET_EHOME_PLAYBACK_OPERATE_MODE  enumMode,
    void                *pOperateParam
)
```

参数

iUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

enumMode

[IN] 控制操作类型，枚举于 [NET_EHOME_PLAYBACK_OPERATE_MODE](#)。

pOperateParam

[IN] 控制操作参数，取决于控制操作类型 (**enumMode**)。关于暂停或恢复回放的参数，详情请参见结构体 [NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StopPlayBack

请求停止回放。

接口定义

```
BOOL CALLBACK NET_ECMS_StopPlayBack (
    LONG    lUserID,
    int     iSessionID
);
```

参数

iUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

iSessionID

[IN] 会话 ID，由 [NET_ECMS_StartPlayBack](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StopPlayBackEx

请求停止回放。该接口支持异步模式。

接口定义

```
BOOL CALLBACK NET_ECMS_StopPlayBackEx (  
    LONG                                iUserID,  
    NET_EHOME_STOPPLAYBACK_PARAM      *pStopParam  
);
```

参数

iUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pStopParam

[IN] 停止回放的请求参数，详情参见结构体 [NET_EHOME_STOPPLAYBACK_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

扩展自

[NET_ECMS_StopPlayBack](#)

14.1.6 语音对讲接口

NET_ECMS_StartVoiceWithStmServer

请求通过流媒体服务器（SMS）开始语音对讲。

接口定义

```
BOOL CALLBACK NET_ECMS_StartVoiceWithStmServer(  
    LONG                lUserID,  
    NET_EHOME_VOICE_TALK_IN *lpVoiceTalkIn  
    NET_EHOME_VOICE_TALK_OUT *lpVoiceTalkOut  
);
```

参数

lUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

lpVoiceTalkIn

[IN] 语音对讲请求的输入参数，详情参见结构体 [NET_EHOME_VOICE_TALK_IN](#)。

lpVoiceTalkOut

[OUT] 语音对讲请求的输出参数，详情参见结构体 [NET_EHOME_VOICE_TALK_OUT](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

参见

[NET_ECMS_StopVoiceTalkWithStmServer](#)

NET_ECMS_StartPushVoiceStream

中心管理服务器（CMS）向设备发送请求，设备开始传输语音对讲的音频码流。

接口定义

```
BOOL CALLBACK NET_ECMS_StartPushVoiceStream(  
    LONG                lUserID,  
    NET_EHOME_PUSHVOICE_IN *lpPushParamIn,  
    NET_EHOME_PUSHVOICE_OUT *lpPushParamOut  
);
```

参数

IUserID

[IN] 用户 ID，由 NET_ECMS_StartListen 注册的回调函数返回。

lpPushParamIn

[IN] 音频码流传输请求的输入参数，详情参见结构体 NET_EHOME_PUSHVOICE_IN。

lpPushParamOut

[OUT] 音频码流传输请求的输出参数，详情参见结构体 NET_EHOME_PUSHVOICE_OUT。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

备注

对于支持 4.0 版本及以上 ISUP 的设备，在调用该接口之前，请确保 CMS 已接收语音对讲请求响应中的设备会话 ID。

参见

NET_ECMS_StartVoiceWithStmServer

NET_ECMS_StopVoiceTalkWithStmServer

请求停止通过流媒体服务器（SMS）进行语音对讲。

接口定义

```
BOOL CALLBACK NET_ECMS_StopVoiceTalkWithStmServer(  
    LONG    lUserID,  
    LONG    lSessionID  
);
```

参数

IUserID

[IN] 用户 ID，由 NET_ECMS_StartListen 注册的回调函数返回。

ISessionID

[IN] 会话 ID，由 NET_ECMS_StartVoiceWithStmServer 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StopVoiceTalkWithStmServerEx

请求停止通过流媒体服务器（SMS）进行语音对讲。该接口支持异步模式。

接口定义

```
BOOL CALLBACK NET_ECMS_StopVoiceTalkWithStmServerEx(  
    LONG                                lUserID,  
    NET_EHOME_STOPVOICETALK_STM_PARAM  *pStopParam  
);
```

参数

lUserID

[IN] 用户 ID

pStopParam

[IN] 停止通过 SMS 进行语音对讲的请求参数，详情参见结构体

[NET_EHOME_STOPVOICETALK_STM_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

扩展自

[NET_ECMS_StopVoiceTalkWithStmServer](#)

NET_ECMS_StartVoiceTalk

开始语音对讲或转发音频。

接口定义

```
LONG CALLBACK NET_ECMS_StartVoiceTalk(  
    LONG                                iUserID,  
    DWORD                               dwVoiceChan,  
    const NET_EHOME_VOICETALK_PARA     *pVoiceTalkPara  
);
```

参数

iUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

dwVoiceChan

[IN] 语音对讲通道号。

pVoiceTalkPara

[IN] 语音对讲或音频转发的参数，详情参见结构体 [NET_EHOME_VOICETALK_PARA](#)。

返回值

返回-1 表示失败，返回其他值表示 [NET_ECMS_StopVoiceTalk](#) 的句柄。

如果返回-1，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

对于 Windows 7 操作系统，如果没有外部音频设备，该接口将返回-1。

参见

[NET_ECMS_SendVoiceTransData](#)

NET_ECMS_SendVoiceTransData

将音频数据转发到设备。

接口定义

```
BOOL CALLBACK NET_ECMS_SendVoiceTransData(  
    LONG        iVoiceHandle,  
    char        *pSendBuf,  
    DWORD       dwBufSize  
);
```

参数

iVoiceHandle

[IN] 音频转发句柄，由 [NET_ECMS_StartVoiceTalk](#) 返回。

pSendBuf

[IN] 保存音频数据的缓冲区的指针。

dwBufSize

[IN] 缓冲区中的数据大小。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_StopVoiceTalk

停止语音对讲或转发音频。

接口定义

```
BOOL CALLBACK NET_ECMS_StopVoiceTalk(  
    LONG    iVoiceHandle  
);
```

参数

iVoiceHandle

[IN] 语音对讲或音频转发的句柄，由 [NET_ECMS_StartVoiceTalk](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

14.1.7 报文传输接口

NET_ECMS_XMLConfig

传输 ISUP 命令和报文。

接口定义

```
BOOL CALLBACK NET_ECMS_XMLConfig(  
    LONG                lUserID,  
    NET_EHOME_XML_CFG  *pXmlCfg,  
    DWORD               dwConfigSize  
);
```

参数

lUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

pXmlCfg

[IN] 被传输的配置参数，不同的配置命令对应于不同的配置参数，详情参见结构体 [NET_EHOME_XML_CFG](#)。

dwConfigSize

[IN] [pXmlCfg](#) 指向的结构体的大小。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

在 ISUP 命令和报文传输过程中，数据只会被发送和接收，不会对数据进行校验。

NET_ECMS_XMLConfigEx

设置 ISAPI（智能安全接入 API）请求 URL 及报文的传输参数。该接口支持异步回调模式。

接口定义

```
BOOL CALLBACK NET_ECMS_XMLConfigEx(  
    LONG                lUserID,  
    NET_EHOME_XML_CFG  *pXmlCfg,  
    DWORD               *pHandle  
)
```

参数

lUserID

[IN] 用户 ID。

pXmlCfg

[IN] ISAPI 请求 URL 及报文的传输参数，详情参见结构体 [NET_EHOME_XML_CFG](#)。

pHandle

[OUT] 异步回调句柄，对应于 [NET_ECMS_SetXmlConfigResponseCB](#) 注册的回调函数 [ASYNC_RESPONSE_CB](#) 中的 **dwHandle**。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

扩展自

[NET_ECMS_XMLConfig](#)

NET_ECMS_XMLRemoteControl

传输 ISUP (智能安全上行协议) 的控制命令和 XML 报文。

接口定义

```
BOOL CALLBACK NET_ECMS_XMLRemoteControl(  
    LONG                                lUserID,  
    NET_EHOME_XML_REMOTE_CTRL_PARAM    *lpCtrlParam,  
    DWORD                               dwCtrlSize  
);
```

参数

lUserID

[IN] 用户 ID, 由 *NET_ECMS_StartListen* 注册的注册回调函数 (*DEVICE_REGISTER_CB*) 返回。

lpCtrlParam

[IN] 控制参数, 详情参见 *NET_EHOME_XML_REMOTE_CTRL_PARAM*。

dwCtrlSize

[IN] 保存控制参数的缓冲区的大小。该缓冲区的大小等于结构体 *NET_EHOME_XML_REMOTE_CTRL_PARAM* 的大小。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 *NET_ECMS_GetLastError* 获取错误码。

备注

该接口不支持传输获取或查询命令。

NET_ECMS_DeletePTXMLConfig

传输请求 URL 和 DELETE 方法。

接口定义

```
BOOL CALLBACK NET_ECMS_DeletePTXMLConfig(  
    LONG                                iUserID,  
    NET_EHOME_PTXML_PARAM              *lpPTXMLParam  
);
```

参数

iUserID

[IN] 用户 ID，由 NET_ECMS_StartListen 注册的回调函数返回。

lpPTXMLParam

[IN] 进行传输的参数，详见结构体 NET_EHOME_PTXML_PARAM。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

备注

该接口只接收或发送数据，不会对数据进行校验或修改，因此必须自行保证数据的正确性。

NET_ECMS_GetPTXMLConfig

传输请求 URL 和 GET 方法。

接口定义

```
BOOL CALLBACK NET_ECMS_GetPTXMLConfig(  
    LONG                iUserID,  
    NET_EHOME_PTXML_PARAM *lpPTXMLParam  
);
```

参数

iUserID

[IN] 用户 ID，由 NET_ECMS_StartListen 注册的回调函数返回。

lpPTXMLParam

[IN] 进行传输的参数，详见结构体 NET_EHOME_PTXML_PARAM。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

备注

该接口只接收或发送数据，不会对数据进行校验或修改，因此必须自行保证数据的正确性。

NET_ECMS_PostPTXMLConfig

传输请求 URL 和 POST 方法。

接口定义

```
BOOL CALLBACK NET_ECMS_PostPTXMLConfig(  
    LONG                iUserID,  
    NET_EHOME_PTXML_PARAM *lpPTXMLParam  
);
```

参数

iUserID

[IN] 用户 ID，由 NET_ECMS_StartListen 注册的回调函数返回。

lpPTXMLParam

[IN] 进行传输的参数，详见结构体 NET_EHOME_PTXML_PARAM。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

备注

该接口只接收或发送数据，不会对数据进行校验或修改，因此必须自行保证数据的正确性。

NET_ECMS_PutPTXMLConfig

传输请求 URL 和 PUT 方法。

接口定义

```
BOOL CALLBACK NET_ECMS_PutPTXMLConfig(  
    LONG                iUserID,  
    NET_EHOME_PTXML_PARAM *lpPTXMLParam  
);
```

参数

iUserID

[IN] 用户 ID，由 NET_ECMS_StartListen 注册的回调函数返回。

lpPTXMLParam

[IN] 进行传输的参数，详见结构体 NET_EHOME_PTXML_PARAM。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

该接口只接收或发送数据，不会对数据进行校验或修改，因此必须自行保证数据的正确性。

NET_ECMS_HTTPConfig

传输 HTTP 页面。

接口定义

```
BOOL CALLBACK NET_ECMS_HTTPConfig (
    LONG                iUserID,
    LPNET_EHOME_HTTP_PARAM lpParam
)
```

参数

iUserID

[IN] 用户 ID，由 [NET_ECMS_StartListen](#) 注册的回调函数返回。

lpParam

HTTP 页面传输参数，参见 [NET_EHOME_HTTP_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。若返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

通过 GET /ISAPI/System/capabilities 返回的报文 XML_DeviceCap 中的节点 <isSupportISUPHttpPassthrough>判断设备是否支持 ISUPV5.0HTTP 页面透传。

NET_ECMS_ISAPIPassThrough

传输命令，包括请求 URL 和操作方法（如 GET、PUT、POST、DELETE）。

接口定义

```
BOOL CALLBACK NET_ECMS_ISAPIPassThrough (
    LONG                lUserID,
```

```
NET_EHOME_PTXML_PARAM *lpParam  
) ;
```

参数

IUserID

[IN] 用户 ID，由 [NET_ECMS StartListen](#) 注册的回调函数返回。

lpParam

[IN] 被传输的参数，详情参见结构体 [NET_EHOME_PTXML_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS GetLastError](#) 获取错误码。

NET_ECMS_SendPassthroughData

使用 HTTP 或私有协议传输数据。

接口定义

```
BOOL CALLBACK NET_ECMS_SendPassthroughData (  
    void* pDataBuffer,  
    DWORD dwDataLen,  
    DWORD dwProxyType  
)
```

参数

pDataBuffer

[IN] 保存要传输的数据的缓冲区。

dwDataLen

[IN] 要传输的数据的大小。

dwProxyType

[IN] 代理类型，默认为“[ENUM_PROXY_TYPE_NETSDK](#)”（私有协议代理）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS GetLastError](#) 获取错误码。

NET_ECMS_SetPassthroughDataCallback

注册回调函数以接收传输数据（通过 HTTP 或私有协议）。

接口定义

```
BOOL CALLBACK NET_ECMS_SetPassthroughDataCallback(  
    PASSTHROUGHDATACALLBACK fnPassthroughDataCb,  
    void *pUser,  
    DWORD dwProxyType  
)
```

参数

fnPassthroughDataCb

[IN] 回调函数指针，详情参见 [PASSTHROUGHDATACALLBACK](#)。

pUser

[IN] 用户指针。

dwProxyType

[IN] 代理类型，默认为“ENUM_PROXY_TYPE_NETSDK”（私有协议代理）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_SetXmlConfigResponseCB

注册用于传输 ISAPI 报文的响应回调函数。

接口定义

```
BOOL CALLBACK NET_ECMS_SetXmlConfigResponseCB(  
    ASYNC_RESPONSE_CB fnCB,  
    void *pUser  
)
```

参数

fnCB

[IN] 响应回调函数，详情参见 [ASYNC_RESPONSE_CB](#)。

pUser

[IN] 用户参数指针。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ECMS_ConvertProtocolHttpToPassthrough

在 HTTP 协议和私有协议之间转换类型。

接口定义

```
LONG CALLBACK NET_ECMS_ConvertProtocolHttpToPassthrough(  
    void *pSrcBuffer,  
    DWORD dwSrcBufLen,  
    void *pDestBuffer,  
    DWORD dwDestBufLen,  
    NET_EHOME_PASSTHROUGH_PARAM *lpParam,  
    BOOL bToPassthrough  
)
```

参数

pSrcBuffer

[IN] 输入缓冲区。

dwSrcBufLen

[IN] 输入缓冲区大小。

pDestBuffer

[IN] 输出缓冲区。

dwDestBufLen

[IN] 输出缓冲区的大小。

lpParam

[IN] 协议参数，详情参见结构体 [NET_EHOME_PASSTHROUGH_PARAM](#)。

bToPassthrough

[IN] 协议转换方向。如果该参数的值为“TRUE”，HTTP 协议开始转换为私有协议以便传输；否则，私有协议开始转换为 HTTP 协议以便传输。

返回值

返回 -1 表示失败，返回其他值表示转换后的数据大小。

如果返回-1，请调用 `NET_ECMS_GetLastError` 获取错误码。

14.2 报警监听模块

14.2.1 基础接口

NET_EALARM_Init

初始化报警管理服务器（AMS）的报警库。

接口定义

```
BOOL CALLBACK NET_EALARM_Init(  
);
```

返回值

返回 `TRUE` 表示成功，返回 `FALSE` 表示失败。

备注

- 如果初始化失败，请检查加载库文件的路径和系统环境。
- 该接口必须与 `NET_EALARM_Fini` 配套调用。

NET_EALARM_Fini

反初始化报警库并释放报警管理服务器（AMS）占用的资源。

接口定义

```
BOOL CALLBACK NET_EALARM_Fini(  
);
```

返回值

返回 `TRUE` 表示成功，返回 `FALSE` 表示失败。

备注

该接口必须与 `NET_EALARM_Init` 配套调用。

NET_EALARM_SetSDKInitCfg

设置报警管理服务器（AMS）的初始化参数。

接口定义

```

BOOL CALLBACK NET_EALARM_SetSDKInitCfg(
    NET_EHOME_EALARM_INIT_CFG_TYPE    enumType,
    void* const                          lpInBuff
)

```

参数

enumType

[IN] 初始化配置类型，详见枚举 [NET_EHOME_EALARM_INIT_CFG_TYPE](#)。

lpInBuff

[IN] 初始化配置参数，取决于配置类型（**enumType**），参见对应关系如下。

enumType	lpInBuff
NET_EHOME_EALARM_INIT_CFG_LIBEAY_PATH	带有 OpenSSL 加密库库名的路径。
NET_EHOME_EALARM_INIT_CFG_SSLEAY_PATH	带有 OpenSSL 通信库库名的路径。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

- 该接口必须在初始化之前调用，否则，设置将不会生效。
- 如果调用失败，无法获取错误码或获取正确的错误码，因为错误码必须在初始化之后才能获取。
- 该接口不会检验 OpenSSL 路径的有效性。加载库时会验证该路径，如果路径不正确，其他相关接口不会报错，错误会写入日志。
- 多个模块用于同一进程时，建议为所有模块设置相同的 OpenSSL 路径。例如，如果注册模块（CMS）以及报警管理模块（AMS）用于同一进程，如果 CMS 已加载 OpenSSL，但 AMS

的 OpenSSL 路径与 CMS 的 OpenSSL 路径不同，且库的版本号也不同，AMS 将使用已加载的 OpenSSL。

- 不同模块（AMS/CMS/SS/SMS）位于统一进程中，加载 OpenSSL 库时，应保证不同模块使用的 OpenSSL 库路径有效，且 OpenSSL 库版本和库名字相同。

NET_EALARM_GetBuildVersion

获取报警管理服务器（AMS）的版本信息。

接口定义

```
DWORD CALLBACK NET_EALARM_GetBuildVersion(  
);
```

返回值

返回版本号，高两位字节表示主版本，低两位字节表示次版本，例如：0x02040001（2.4.0.1 版本）。

NET_EALARM_SetLogToFile

设置参数以启用报警管理服务器（AMS）的日志功能。

接口定义

```
BOOL CALLBACK NET_EALARM_SetLogToFile(  
    LONG        nLogLevel,  
    char        *strLogDir,  
    BOOL        bAutoDel  
);
```

参数

nLogLevel

[IN] 日志类型。0-启用日志功能（默认），1-错误日志，2-错误和调试日志，3-错误、调试和信息日志。

strLogDir

[IN] 日志文件保存路径。Windows 操作系统中的默认保存路径为“C:\\SdkLog\\”，Linux 操作系统中的默认保存路径为“/home/sdklog/”。如果要使用默认保存路径，请将该参数设置为“NULL”。

bAutoDel

[IN] 是否在一段时间后自动删除日志文件：“TRUE”-是（默认），“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩（将这生成的 50 个日志文件压缩成一个压缩文件）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_EALARM_GetLastError](#) 获取错误码。

备注

- 日志文件的保存路径必须是绝对路径，且必须以“\”结尾，例如：“C:\SdkLog\”。
- 如果更改了保存路径，更改后的路径将用于保存下一个日志文件。
- 当 **bAutoDel** 的值为“TRUE”时，第一个日志文件将不会被删除，因为该文件包含起始时间、版本信息和其他重要信息。
- 允许 HCISUPSDK 在运行时，进行日志参数的修改。可在运行目录下添加 HCISUPSDK_Log_Switch.xml 文件来实现运行时日志参数的配置，可参见 [日志参数配置](#)。

NET_EALARM_GetLastError

如果调用失败或完成，可获取错误码。

接口定义

```
DWORD CALLBACK NET_EALARM_GetLastError(  
);
```

返回值

返回值是错误码，详情参见 [ISUPSDK 错误码](#)。

NET_EALARM_GetSDKLocalCfg

获取报警管理服务器（AMS）的本地配置参数。

接口定义

```
BOOL CALLBACK NET_EALARM_GetSDKLocalCfg(  
    NET_EHOME_LOCAL_CFG_TYPE    enumType,  
    void*                        lpOutBuff  
);
```

参数

enumType

[IN] 支持的本地配置类型宏定义值：0, 4 和 5, 参见枚举定义 [NET_EHOME_LOCAL_CFG_TYPE](#)

。

IpOutBuff

[OUT] 本地配置参数，与本地配置类型（`enumType`）有关，详情参见下表。

enumType	IpOutBuff	说明
UNDEFINE	未定义	未定义。
ACTIVE_ACCESS_SECURITY	NET_EHOME_LOCAL_ACCESS_SECURITY	获取 AMS 模块设备主动接入的安全性信息。
LOCAL_CFG_TYPE_GENERAL	NET_EHOME_LOCAL_GENERAL_CFG	获取 AMS 模块本地通用配置。
COM_PATH	COM 路径，指向一个字符串缓冲区（例如：“./HCApSDKCom/”）。	获取 AMS 模块的 HCApSDKCom（为 HCISUP 相关依赖库的目录）路径。

返回值

返回 `TRUE` 表示成功，返回 `FALSE` 表示失败。

如果返回 `FALSE`，请调用 [NET_EALARM_GetLastError](#) 获取错误码。

参见

[NET_EALARM_SetSDKLocalCfg](#)

NET_EALARM_SetSDKLocalCfg

设置报警管理服务器（AMS）的本地配置参数。

接口定义

```

BOOL CALLBACK NET_EALARM_SetSDKLocalCfg(
    NET_EHOME_LOCAL_CFG_TYPE    enumType,
    void* const                  lpInBuff
);

```

参数

`enumType`

[IN] 支持的本地配置类型宏定义值：-1, 0, 1, 2, 3 和 4, 参见枚举定义 NET_EHOME_LOCAL_CFG_TYPE。

IpInBuff

[IN] 本地配置参数，与本地配置类型（enumType）相关，详情参见下表。

enumType	IpOutBuff	说明
UNDEFINE	未定义	未定义。
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u>	设置 AMS 模块设备主动接入的安全性信息。
LOCAL_CFG_TYPE_GENERAL	<u>NET_EHOME_LOCAL_GENERAL_CFG</u>	设置 AMS 模块本地通用配置。
COM_PATH	COM 路径，指向一个字符串缓冲区（例如：“./HCApSDKCom/”）。	设置 AMS 模块的 HCApSDKCom（为 HCISUP 相关依赖库的目录）路径。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_EALARM_GetLastError 获取错误码。

参见

NET_EALARM_GetSDKLocalCfg

14.2.2 配置和控制接口

NET_EALARM_GetDeviceSessionKey

获取设备的会话密钥信息。

接口定义

```

BOOL CALLBACK NET_EALARM_GetDeviceSessionKey(
    NET_EHOME_DEV_SESSIONKEY *pDeviceKey
)

```

参数

pDeviceKey

[OUT] 会话密钥信息，详情参见 [NET_EHOME_DEV_SESSIONKEY](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_EALARM_GetLastError](#) 获取错误码。

参见

[NET_EALARM_SetDeviceSessionKey](#)

NET_EALARM_SetDeviceSessionKey

为设备设置会话密钥。

接口定义

```
BOOL CALLBACK NET_EALARM_SetDeviceSessionKey(  
    NET_EHOME_DEV_SESSIONKEY *deviceKey  
);
```

参数

deviceKey

[IN] 支持 5.0 版本 ISUP 设备的会话密钥信息，详情参见 [NET_EHOME_DEV_SESSIONKEY](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_EALARM_GetLastError](#) 获取错误码。

备注

- 用于 AMS 模块对 ISUP5.0 协议设备认证时设置 SessionKey。
- 设备的 SessionKey 由 CMS 模块的注册回调 ENUM_DEV_SESSIONKEY 时产生。需要通知 AMS 模块设备使用的 SessionKey，AMS 模块通过调用该接口来设置设备认证使用的 SessionKey。设置了正确的 SessionKey，ISUP5.0 协议设备才可以正常接入 AMS 并推送报警信息。

参见

[NET_EALARM_GetDeviceSessionKey](#)

14.2.3 监听接口

NET_EALARM_StartListen

启用报警管理服务器（AMS）监听并注册回调函数以接收报警信息。

接口定义

```
LONG CALLBACK NET_EALARM_StartListen(  
    const NET_EHOME_ALARM_LISTEN_PARAM    *pAlarmListenParam  
);
```

参数

pAlarmListenParam

[IN] 开启监听后用于接收报警信息的输入参数，详情参见结构体 *NET_EHOME_ALARM_LISTEN_PARAM*。

返回值

返回-1表示失败，返回其他值表示 *NET_EALARM_StopListen* 的句柄。如果返回-1，请调用 *NET_EALARM_GetLastError* 获取错误码。

NET_EALARM_StopListen

停止报警管理服务器（AMS）的监听。

接口定义

```
BOOL CALLBACK NET_EALARM_StopListen(  
    LONG    lListenHandle  
);
```

参数

lListenHandle

[IN] 报警监听句柄，由 *NET_EALARM_StartListen* 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。如果返回 *FALSE*，请调用 *NET_EALARM_GetLastError* 获取错误码。

14.3 取流模块

14.3.1 基础接口

NET_ESTREAM_Init

初始化取流模块库。

接口定义

```
BOOL CALLBACK NET_ESTREAM_Init(  
);
```

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

该接口必须与 [NET_ESTREAM_Fini](#) 配套调用。

NET_ESTREAM_Fini

释放取流模块占用的资源。

接口定义

```
BOOL CALLBACK NET_ESTREAM_Fini(  
);
```

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

该接口必须与 [NET_ESTREAM_Init](#) 配套调用。

NET_ESTREAM_SetSDKInitCfg

设置流媒体服务器（SMS）初始化参数。

接口定义

```

BOOL CALLBACK NET_ESTREAM_SetSDKInitCfg (
    NET_EHOME_ESTREAM_INIT_CFG_TYPE  enumType,
    void* const                        lpInBuff
)

```

参数

enumType

[IN] 初始化配置类型，详见枚举 [NET_EHOME_ESTREAM_INIT_CFG_TYPE](#)。

lpInBuff

[IN] 初始化参数，取决于配置类型，参见详情如下。

enumType	lpInBuff
NET_EHOME_ESTREAM_INIT_CFG_LIBEAY_PATH	带有 OpenSSL 加密库库名的路径。
NET_EHOME_ESTREAM_INIT_CFG_SSLEAY_PATH	带有 OpenSSL 通信库库名的路径。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

- 该接口必须在初始化之前调用，否则，设置将不会生效。
- 如果调用失败，无法获取错误码或获取正确的错误码，因为错误码必须在初始化之后才能获取。
- 该接口不会检验 OpenSSL 路径的有效性。加载库时会验证该路径，如果路径不正确，其他相关接口不会报错，错误会写入日志。
- 多个模块用于同一进程时，建议为所有模块设置相同的 OpenSSL 路径。例如，如果注册模块（CMS）以及报警管理模块（AMS）用于同一进程，如果 CMS 已加载 OpenSSL，但 AMS 的 OpenSSL 路径与 CMS 的 OpenSSL 路径不同，且库的版本号也不同，AMS 将使用已加载的 OpenSSL。
- 不同模块（AMS/CMS/SS/SMS）位于统一进程中，加载 OpenSSL 库时，应保证不同模块使用的 OpenSSL 库路径有效，且 OpenSSL 库版本和库名字相同。

NET_ESTREAM_GetBuildVersion

获取流媒体服务器（SMS）的版本信息。

接口定义

```
DWORD CALLBACK NET_ESTREAM_GetBuildVersion(  
);
```

返回值

返回版本号，高两位字节表示主版本，低两位字节表示次版本，例如：0x02040001（2.4.0.1版本）。

NET_ESTREAM_SetLogToFile

设置启用流媒体服务器（SMS）日志功能的参数。

接口定义

```
BOOL CALLBACK NET_ESTREAM_SetLogToFile(  
    DWORD    nLogLevel,  
    char     *strLogDir,  
    BOOL     bAutoDel  
);
```

参数

nLogLevel

[IN] 日志类型：0-启用日志功能（默认），1-错误日志，2-错误和调试日志，3-错误、调试和信息日志。

strLogDir

[IN] 日志文件保存路径。Windows 操作系统中的默认保存路径为“C:\\SdkLog\\”，Linux 操作系统中的默认保存路径为“/home/sdklog/”。如果要使用默认保存路径，请将该参数设置为“NULL”。

bAutoDel

[IN] 是否在一段时间后自动删除日志文件：“TRUE”-是（默认），“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩（将这生成的 50 个日志文件压缩成一个压缩文件）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESTREAM_GetLastError](#) 获取错误码。

备注

- 日志文件的保存路径必须是绝对路径，且必须以“\\”结尾，例如：“C:\\SdkLog\\”。
- 如果更改了保存路径，更改后的路径将用于保存下一个日志文件。
- 当 **bAutoDel** 的值为“TRUE”时，第一个日志文件将不会被删除，因为该文件包含起始时间、版本信息和其他重要信息。
- 允许 HCISUPSDK 在运行时，进行日志参数的修改。可在运行目录下添加 HCISUPSDK_Log_Switch.xml 文件来实现运行时日志参数的配置，可参见 [日志参数配置](#)。

NET_ESTREAM_GetLastError

若取流失败或完成，获取错误码。

接口定义

```
DWORD CALLBACK NET_ESTREAM_GetLastError(  
);
```

返回值

返回值是错误码，详情参见 [ISUPSDK 错误码](#)。

NET_ESTREAM_GetSDKLocalCfg

获取流媒体服务器（SMS）的本地配置参数。

接口定义

```
BOOL CALLBACK NET_ESTREAM_GetSDKLocalCfg(  
    NET_EHOME_LOCAL_CFG_TYPE    enumType,  
    void                          *lpOutBuff  
);
```

参数

enumType

[IN] 支持的本地配置类型宏定义值：0 和 5，参见枚举定义 [NET_EHOME_LOCAL_CFG_TYPE](#)。

lpOutBuff

[OUT] 本地配置参数，与本地配置类型（**enumType**）有关，详情参见下表。

enumType	lpOutBuff
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u>
COM_PATH	COM 路径。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ESTREAM_GetLastError 获取错误码。

参见

NET_ESTREAM_SetSDKLocalCfg

NET_ESTREAM_SetSDKLocalCfg

设置流媒体服务器（SMS）的本地配置参数。

接口定义

```

BOOL CALLBACK NET_ESTREAM_SetSDKLocalCfg(
    NET_EHOME_LOCAL_CFG_TYPE    enumType,
    void* const                  lpInBuff
);

```

参数

enumType

[IN] 支持的本地配置类型宏定义值：0，参见枚举定义 NET_EHOME_LOCAL_CFG_TYPE。

lpInBuff

[IN] 本地配置参数，与本地配置类型（**enumType**）相关，详情参见下表。

enumType	lpOutBuff	说明
UNDEFINE	未定义。	未定义。
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u>	设置 SMS 模块的设备主动接入安全性。

enumType	lpOutBuff	说明
COM_PATH	COM 路径，指向一个字符串缓冲区。	获取 SMS 模块的 HCApSDKCom（为 HCISUP 相关依赖库的目录）的路径。
STREAM_PLAYBACK_PARAM	<u>NET_EHOME_LOCAL_PLAYBACK_PARAM</u>	设置是否开启回放同步接收。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ESTREAM_GetLastError 获取错误码。

备注

NET_ESTREAM_GetSDKLocalCfg

NET_ESTREAM_SetExceptionCallback

注册回调函数以接收异常信息。

接口定义

```

BOOL CALLBACK NET_ESTREAM_SetExceptionCallback(
    DWORD                dwMessage,
    HANDLE               hWnd,
    fExceptionCallback  cbExceptionCallback,
    void                 *pUser
);

```

参数

dwMessage

[IN] 保留。

hWnd

[IN] 用于接收异常信息的窗口句柄。

cbExceptionCallback

[IN] 接收异常信息的回调函数，详情参见 fExceptionCallback。

pUser

[IN] 用户参数。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

14.3.2 预览接口

NET_ESTREAM_StartListenPreview

启动流媒体服务器（SMS）的监听并注册回调函数以接收设备连接请求。

接口定义

```
LONG CALLBACK NET_ESTREAM_StartListenPreview(  
    NET_EHOME_LISTEN_PREVIEW_CFG      *pListenParam  
);
```

参数

pListenParam

[IN] 预览监听参数，详情参见结构体 [NET_EHOME_LISTEN_PREVIEW_CFG](#)。

返回值

返回-1表示失败，返回其他值表示 [NET_ESTREAM_StopListenPreview](#) 的句柄。

如果返回-1，请调用 [NET_ESTREAM_StopListenPreview](#) 获取错误码。

NET_ESTREAM_StopListenPreview

停止流媒体服务器（SMS）的预览监听并断开与设备的连接。

接口定义

```
BOOL CALLBACK NET_ESTREAM_StopListenPreview(  
    LONG      lListenHandle  
);
```

参数

lListenHandle

[IN] 预览监听句柄，由 [NET_ESTREAM_StartListenPreview](#) 返回。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

调用该接口后, 该监听端口的所有链接将被断开。

NET_ESTREAM_StopPreview

停止某一通道转发预览实时码流。

接口定义

```
BOOL CALLBACK NET_ESTREAM_StopPreview(  
    LONG      lPreviewHandle  
);
```

参数

lPreviewHandle

[IN] 预览监听句柄, 由 [NET_ESTREAM_StartListenPreview](#) 注册的回调函数 [PREVIEW_NEWLINK_CB](#) 返回。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

对于与多个通道相关联的监听端口, 调用该接口后, 仅其中一个通道的预览将会停止。

NET_ESTREAM_SetPreviewDataCB

注册回调函数以接收实时码流。

接口定义

```
BOOL CALLBACK NET_ESTREAM_SetPreviewDataCB(  
    LONG      lHandle,  
    NET_EHOME_PREVIEW_DATA_CB_PARAM *pStruCBParam  
);
```

参数

IHandle

[IN] 预览句柄，由 NET_ESTREAM_StartListenPreview 注册的回调函数 PREVIEW_NEWLINK_CB 返回。

pStruCBParam

[IN] 预览数据的回调参数，详情参见结构体 NET_EHOME_PREVIEW_DATA_CB_PARAM。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ECMS_GetLastError 获取错误码。

NET_ESTREAM_SetStandardPreviewDataCB

注册预览（RTP 码流）回调函数。

接口定义

```
BOOL CALLBACK NET_ESTREAM_SetStandardPreviewDataCB(  
    LONG                                iHandle,  
    NET_EHOME_PREVIEW_DATA_CB_PARAM    *pStruCBParam  
)
```

参数

iHandle

[IN] 预览监听句柄，由 NET_ESTREAM_StartListenPreview 返回。

pStruCBParam

[IN] 被回调的预览参数，详情参见 NET_EHOME_PREVIEW_DATA_CB_PARAM。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 NET_ESTREAM_GetLastError 获取错误码。

14.3.3 回放接口

NET_ESTREAM_SetPlayBackDataCB

注册回调函数以接收回放码流。

接口定义

```
BOOL CALLBACK NET_ESTREAM_SetPlayBackDataCB (  
    LONG  
    NET_EHOME_PLAYBACK_DATA_CB_PARAM    *pDataCBParam  
);
```

参数

iPlayBackLinkHandle

[IN] 回放句柄，由 *NET_ESTREAM_StartListenPlayBack* 注册的回调函数 *PLAYBACK_NEWLINK_CB* 返回。

pDataCBParam

[IN] 回放数据的回调参数，详情参见结构体 *NET_EHOME_PLAYBACK_DATA_CB_PARAM*。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 *NET_ECMS_GetLastError* 获取错误码。

NET_ESTREAM_StartListenPlayBack

启用流媒体服务器（SMS）的回放监听并注册回调函数以接收设备连接请求。

接口定义

```
LONG CALLBACK NET_ESTREAM_StartListenPlayBack (  
    NET_EHOME_PLAYBACK_LISTEN_PARAM    *pListenParam  
);
```

参数

pListenParam

[IN] 回放监听参数，详情参见结构体 *NET_EHOME_PLAYBACK_LISTEN_PARAM*。

返回值

返回-1表示失败，返回其他值表示 *NET_ESTREAM_StopListenPlayBack* 的句柄。

如果返回-1，请调用 *NET_ESTREAM_GetLastError* 获取错误码。

NET_ESTREAM_StopListenPlayBack

停止流媒体服务器（SMS）的回放监听并断开与设备的连接。

接口定义

```
BOOL CALLBACK NET_ESTREAM_StopListenPlayBack(  
    LONG    iPlaybackListenHandle  
);
```

参数

iPlaybackListenHandle

[IN] 回放的监听句柄, 由 [NET_ESTREAM_StartListenPlayBack](#) 返回。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

调用该接口后, 该监听端口的所有链接将被断开。

NET_ESTREAM_StopPlayBack

停止某一通道转发回放码流。

接口定义

```
BOOL CALLBACK NET_ESTREAM_StopPlayBack(  
    LONG    iPlayBackLinkHandle  
);
```

参数

iPlayBackLinkHandle

[IN] 回放监听句柄, 由 [NET_ESTREAM_StartListenPlayBack](#) 注册的回调函数

[PLAYBACK_NEWLINK_CB](#) 返回。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

对于与多个通道相关联的监听端口,调用该接口后,仅其中一个通道的回放将会停止。

14.3.4 语音对讲接口

NET_ESTREAM_StartListenVoiceTalk

启用流媒体服务器（SMS）的语音对讲监听并注册回调函数以接收设备连接请求。

接口定义

```
LONG CALLBACK NET_ESTREAM_StartListenVoiceTalk(
    NET_EHOME_LISTEN_VOICETALK_CFG      *pListenParam
);
```

参数

pListenParam

[IN] 语音对讲的监听参数，详情参见结构体 [NET_EHOME_LISTEN_VOICETALK_CFG](#)。

返回值

返回-1表示失败，返回其他值表示 [NET_ESTREAM_StopListenVoiceTalk](#) 的句柄。
如果返回-1，请调用 [NET_ESTREAM_GetLastError](#) 获取错误码。

参见

[NET_ESTREAM_SendVoiceTalkData](#)

NET_ESTREAM_SetVoiceTalkDataCB

注册回调函数以接收音频码流。

接口定义

```
BOOL CALLBACK NET_ESTREAM_SetVoiceTalkDataCB(
    LONG                                lHandle,
    NET_EHOME_VOICETALK_DATA_CB_PARAM  *pStruCBParam
);
```

参数

lHandle

[IN] 语音对讲句柄，由 [NET_ESTREAM_StartListenVoiceTalk](#) 注册的回调函数 [VOICETALK_NEWLINK_CB](#) 返回。

pStruCBParam

[IN] 语音对讲数据的回调参数, 详情参见结构体 [NET_EHOME_VOICETALK_DATA_CB_PARAM](#)。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

NET_ESTREAM_SendVoiceTalkData

将音频数据转发到设备。

接口定义

```
LONG CALLBACK NET_ESTREAM_SendVoiceTalkData(  
    LONG                                lHandle,  
    NET_EHOME_VOICETALK_DATA            *pVoicTalkData  
);
```

参数

lHandle

[IN] 语音对讲句柄, 由 [NET_ESTREAM_StartListenVoiceTalk](#) 注册的监听回调函数 ([VOICETALK_NEWLINK_CB](#)) 返回。

pVoicTalkData

[IN] 待转发的音频数据, 详情参见结构体 [NET_EHOME_VOICETALK_DATA](#)。

返回值

返回 *TRUE* 表示成功, 返回 *FALSE* 表示失败。

如果返回 *FALSE*, 请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

- 待转发至设备的音频数据的编码类型与设备本身的语音对讲编码类型相同。
- 对于使用 G.722 编码类型编码的音频, 每次可转发的数据大小为 80 字节; 对于使用 G.711 编码类型编码的音频, 每次可转发的数据大小为 160 字节。

NET_ESTREAM_StopListenVoiceTalk

停止流媒体服务器 (SMS) 的语音对讲监听并断开与设备的连接。

接口定义

```
BOOL CALLBACK NET_ESTREAM_StopListenVoiceTalk(  
    LONG    lListenHandle  
);
```

参数

lListenHandle

[IN] 语音对讲的监听句柄，由 [NET_ESTREAM_StartListenVoiceTalk](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

调用该接口后，该监听端口的所有链接将被断开。

NET_ESTREAM_StopVoiceTalk

停止某一通道转发语音对讲的音频数据。

接口定义

```
BOOL CALLBACK NET_ESTREAM_StopVoiceTalk(  
    LONG    lHandle  
);
```

参数

lHandle

[IN] 语音对讲的监听句柄，由 [NET_ESTREAM_StartListenVoiceTalk](#) 注册的回调函数 [VOICETALK_NEWLINK_CB](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ECMS_GetLastError](#) 获取错误码。

备注

对于与多个通道相关联的监听端口，调用该接口后，仅其中一个通道的语音对讲将会停止。

14.4 存储管理模块

14.4.1 基础接口

NET_ESS_Init

初始化存储管理模块库。

接口定义

```
BOOL CALLBACK NET_ESS_Init(  
);
```

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

- 如果初始化失败，请检查加载库文件的路径和系统环境。
- 该接口必须与 *NET_ESS_Fini* 配套调用。

NET_ESS_Init_V11

设置初始化参数。

接口定义

```
BOOL CALLBACK NET_ESS_Init_V11(  
NET_EHOME_SS_INIT_PARAM *pParam  
);
```

参数

pParam

初始化参数，参见结构体 *NET_EHOME_SS_INIT_PARAM*。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 *NET_ESS_GetLastError* 获取错误码。

NET_ESS_Fini

反初始化存储管理库并释放存储服务器 (SS) 占用的资源。

接口定义

```
BOOL CALLBACK NET_ESS_Fini(  
);
```

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

该接口必须与 *NET_ESS_Init* 配套调用。

NET_ESS_GENERATE_SECRETKEY

计算加密密钥。

接口定义

```
BOOL CALLBACK NET_ESS_GENERATE_SECRETKEY(  
    const char    *pSrc,  
    const char    *pKey,  
    char          *pSecretKeyOut,  
    DWORD        dwSecretKeyLen  
);
```

参数

pSrc

[IN] 输入数据缓冲区。

pKey

[IN] 密钥缓冲区。

pSecretKeyOut

[IN/OUT] 加密密钥缓冲区。

dwSecretKeyLen

[IN] 加密密钥缓冲区大小。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 *NET_ESS_GetLastError* 获取错误码。

NET_ESS_HAMSHA256

计算 HMAC-SHA256 的值。

接口定义

```
BOOL CALLBACK NET_ESS_HAMSHA256(  
    const char*    pSrc,  
    const char*    pSecretKey,  
    char*          pSingatureOut,  
    DWORD          dwSingatureLen  
)
```

参数

pSrc

[IN] 输入数据，一般指向设备 ID。

pSecretKey

[IN] 密钥，一般指向 EHomeKey。

pSingatureOut

[IN/OUT] 目的数据缓冲区。

dwSingatureLen

[IN] 目的数据缓冲区长度。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 *NET_ESS_GetLastError* 获取错误码。

NET_ESS_SetSDKInitCfg

设置存储服务器 (SS) 初始化参数。

接口定义

```
BOOL CALLBACK NET_ESS_SetSDKInitCfg(  
    NET_EHOME_SS_INIT_CFG_TYPE    enumType,  
    void* const                    lpInBuff  
) ;
```

参数

enumType

[IN] 初始化配置类型，详见枚举 [NET_EHOME_SS_INIT_CFG_TYPE](#)。

lpInBuff

[IN] 初始化配置参数，取决于配置类型（enumType），对应关系如下。

enumType	lpInBuff	说明
NET_EHOME_SS_INIT_CFG_SDK_PATH	<u>NET_EHOME_SS_LOCAL_SDK_PATH</u>	设置 SS 组件加载路径（仅 Linux 版本支持）。
NET_EHOME_SS_INIT_CFG_CLOUD_TIME_DIFF	指向一个 DWORD 类型，值为时差，单位：分钟。	设置 SS 模块，云存储的请求时间差值。默认 15 分钟，最小 5 分钟，最大 60 分钟。
NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT	<u>NET_EHOME_IPADDRESS</u>	设置 SS 模块的公网地址（当存在内外网映射时使用）。
NET_EHOME_SS_INIT_CFG_LIBEAY_PATH	OpenSSL 加密库路径（含库名）。示例“./libeay32.dll”。	设置 SS 模块使用 OpenSSL 加密库（Windows 下为 libeay32.dll；Linux 下为 libcrypto.so）路径。
NET_EHOME_SS_INIT_CFG_SSLEAY_PATH	OpenSSL 通信库路径（含库名）。示例“./ssleay32.dll”。	设置 SS 模块使用 OpenSSL 通信库（Windows 下为 ssleay32.dll；Linux 下为 libssl.so）路径。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

备注

- 该接口必须在初始化之前调用，否则，设置将不会生效。
- 如果调用失败，无法获取错误码或获取正确的错误码，因为错误码必须在初始化之后才能获取。
- 该接口不会检验 OpenSSL 路径的有效性。加载库时会验证该路径，如果路径不正确，其他相关接口不会报错，错误会写入日志。

- 多个模块用于同一进程时，建议为所有模块设置相同的 OpenSSL 路径。例如，如果注册模块（CMS）以及报警管理模块（AMS）用于同一进程，如果 CMS 已加载 OpenSSL，但 AMS 的 OpenSSL 路径与 CMS 的 OpenSSL 路径不同，且库的版本号也不同，AMS 将使用已加载的 OpenSSL。
- 不同模块（AMS/CMS/SS/SMS）位于统一进程中，加载 OpenSSL 库时，应保证不同模块使用的 OpenSSL 库路径有效，且 OpenSSL 库版本和库名字相同。

NET_ESS_SetLogToFile

设置启用存储服务器（SS）日志功能的参数。

接口定义

```
BOOL CALLBACK NET_ESS_SetLogToFile(  
    LONG                iLogLevel,  
    const char          *strLogDir,  
    BOOL                bAutoDel  
);
```

参数

iLogLevel

[IN] 日志类型：1-错误日志，2-调试日志，3-信息日志。

strLogDir

[IN] 日志文件保存路径。Windows 操作系统中的默认保存路径为“C:\\SdkLog\\”，Linux 操作系统中的默认保存路径为“/home/sdklog/”。如果要使用默认保存路径，请将该参数设置为“NULL”。

bAutoDel

[IN] 是否在一段时间后自动删除日志文件：“TRUE”-是（默认），“FALSE”-否。当为“FALSE”时，会启动日志压缩，每生成 50 个日志文件时，将会进行一次日志压缩（将这生成的 50 个日志文件压缩成一个压缩文件）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

备注

- 日志文件的保存路径必须是绝对路径，且必须以“\\”结尾，例如：“C:\\SdkLog\\”。
- 如果更改了保存路径，更改后的路径将用于保存下一个日志文件。

- 当 **bAutoDel** 的值为“TRUE”时，第一个日志文件将不会被删除，因为该文件包含起始时间、版本信息和其他重要信息。
- 允许 HCISUPSDK 在运行时，进行日志参数的修改。可在运行目录下添加 HCISUPSDK_Log_Switch.xml 文件来实现运行时日志参数的配置，可参见 [日志参数配置](#)。

NET_ESS_GetBuildVersion

获取存储服务器（SS）的版本信息。

接口定义

```
DWORD CALLBACK NET_ESS_GetBuildVersion(  
);
```

返回值

返回版本号，高两位字节表示主版本，低两位字节表示次版本，例如：0x02040001（2.4.0.1 版本）。

NET_ESS_GetLastError

如果调用失败或完成，可获取错误码。

接口定义

```
DWORD CALLBACK NET_ESS_GetLastError(  
);
```

返回值

返回值是错误码，详情参见 [ISUPSDK 错误码](#)。

NET_ESS_GenerateStorageUri

生成存储资源（图片等）的 URL 信息。

接口定义

```
BOOL CALLBACK NET_ESS_GenerateStorageUri(  
    NET_EHOME_SS_STORAGE_URI* pSSStorageUri  
);
```

参数

pSSStorageUri

存储资源的 URI 信息，指向一个 [NET_EHOME_SS_STORAGE_URI](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

14.4.2 监听接口

NET_ESS_StartListen

启用存储服务器（SS）的监听并注册回调函数以接收存储的文件。

接口定义

```
LONG CALLBACK NET_ESS_StartListen(  
    NET_EHOME_SS_LISTEN_PARAM    *pSSListenParam  
);
```

参数

pSSListenParam

[IN] 监听参数，详情参见结构体 [NET_EHOME_SS_LISTEN_PARAM](#)。

返回值

返回 -1 表示失败，返回其他值表示 [NET_ESS_StopListen](#) 的句柄参数。

如果返回 -1，请调用 [NET_ESS_GetLastError](#) 获取错误码。

Remarks

在调用此接口前，需要调用接口 [NET_ESS_SetListenHttpsParam](#) 设置 HTTPS 监听参数；如果不启用 HTTPS，则需要调用接口 [NET_ESS_SetListenHttpsParam](#) 清空参数值，否则设置过的 HTTPS 监听参数仍继续生效。

NET_ESS_StopListen

停止存储服务器（SS）的监听。

接口定义

```
BOOL CALLBACK NET_ESS_StopListen(  
    LONG    iHandle  
);
```

参数

iHandle

[IN] 监听句柄，由 [NET_ESS_StartListen](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

NET_ESS_SetListenHttpsParam

设置 HTTPS 监听参数。

接口定义

```
BOOL CALLBACK NET_ESS_SetListenHttpsParam(  
    NET_EHOME_SS_LISTEN_HTTPS_PARAM *pSSHttpsParam  
)
```

参数

pSSHttpsParam

HTTPS 监听参数，详情参见结构体 [NET_EHOME_SS_LISTEN_HTTPS_PARAM](#)。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

14.4.3 客户端相关接口

NET_ESS_CreateClient

创建客户端以上传文件。

接口定义

```
LONG CALLBACK NET_ESS_CreateClient(  
    NET_EHOME_SS_CLIENT_PARAM *pClientParam  
) ;
```

参数

pClientParam

[IN] 客户端参数，详情参见结构体 [NET_EHOME_SS_CLIENT_PARAM](#)。

返回值

返回值为 LONG 类型，小于 0 时，表示失败；大于等于 0 表示成功，为客户端句柄，用于 [NET_ESS_ClientSetTimeout](#)、[NET_ESS_ClientSetParam](#)、[NET_ESS_ClientDoUpload](#)、[NET_ESS_ClientDoDownload](#)、[NET_ESS_DestroyClient](#) 接口的相关参数。

NET_ESS_ClientSetParam

设置上传文件的客户端参数。

接口定义

```

BOOL CALLBACK NET_ESS_ClientSetParam(
    LONG                lHandle,
    const char          *strParamName,
    const char          *strParamVal
);

```

参数

lHandle

[IN] 客户端句柄，由 [NET_ESS_CreateClient](#) 返回。

strParamName

[IN] 客户端参数名，详情参见下表。

strParamName	描述
SS_CLIENT_FILE_PATH_PARAM_NAME	文件的保存路径。
SS_CLIENT_VRB_FILENAME_CODE	VRB 协议 FilenameCode
SS_CLIENT_KMS_USER_NAME	密钥管理服务器（KMS）的用户名
SS_CLIENT_KMS_PASSWORD	密钥管理服务器（KMS）的密码
SS_CLIENT_CLOUD_AK_NAME	5.0 版本 ISUP 的访问密钥（AccessKey）
SS_CLIENT_CLOUD_SK_NAME	5.0 版本 ISUP 的密钥（SecretKey）

strParamVal

[IN] 客户端参数值，取决于参数名（**strParamName**）。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 *NET_ESS_GetLastError* 获取错误码。

NET_ESS_ClientSetTimeout

设置通过客户端上传或下载文件的超时时间。

接口定义

```
BOOL CALLBACK NET_ESS_ClientSetTimeout(  
    LONG          lHandle,  
    DWORD         dwSendTimeout,  
    DWORD         dwRecvTimeout  
);
```

参数

lHandle

[IN] 客户端句柄，由 *NET_ESS_CreateClient* 返回。

dwSendTimeout

[IN] 文件上传超时时间。

dwRecvTimeout

[IN] 文件下载超时时间。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 *NET_ESS_GetLastError* 获取错误码。

NET_ESS_ClientDoUpload

开始通过客户端上传文件。

接口定义

```
BOOL CALLBACK NET_ESS_ClientDoUpload(  
    LONG          lHandle,  
    char         *strUrl,
```

```
LONG      dwUrlLen  
) ;
```

参数

IHandle

[IN] 客户端句柄，由 *NET_ESS_CreateClient* 返回。

strUrl

[IN] 文件 URL。

dwUrlLen

[IN] 文件 URL 长度。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 *NET_ESS_GetLastError* 获取错误码。

NET_ESS_ClientDoUploadBuffer

客户端通过内存数据的方式上传资源（图片）。

接口定义

```
BOOL CALLBACK NET_ESS_ClientDoUploadBuffer(  
    LONG      lHandle,  
    char*     strUrl,  
    DWORD     dwUrlLen,  
    void*     pFileContent,  
    DWORD     dwContentLen  
)
```

参数

IHandle

[IN] 客户端句柄，*NET_ESS_CreateClient* 的返回值。

strUrl

[IN/OUT] URL 缓冲区。

dwUrlLen

[IN] URL 缓冲区大小。

pFileContent

[IN] 指向资源缓冲区（如图片数据）。

dwContentLen

[IN] 资源缓冲区大小。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

NET_ESS_ClientDoDelete

通过客户端从存储服务器中删除指定文件。

接口定义

```
BOOL NET_ESS_ClientDoDelete(  
    LONG          lHandle,  
    const char    *strUrl  
);
```

参数

lHandle

[IN] 客户端句柄，由 [NET_ESS_CreateClient](#) 返回。

strUrl

[IN] 待删除的指定文件 URL，由 [NET_ESS_ClientDoUpload](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

NET_ESS_ClientDoDownload

开始通过客户端下载文件。

接口定义

```
BOOL CALLBACK NET_ESS_ClientDoDownload(  
    LONG          lHandle,  
    char          *strUrl,  
    void          **pFileContent,  
    DWORD&       dwContentLen  
);
```

参数

IHandle

[IN] 客户端句柄，由 [NET_ESS_CreateClient](#) 返回。

strUrl

[IN] 图片 URL。

pFileContent

[OUT] 接收下载数据的缓冲区。

dwContentLen

[OUT] 接收缓冲区中数据的大小。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

NET_ESS_DestroyClient

销毁客户端。

接口定义

```
BOOL CALLBACK NET_ESS_DestroyClient(  
    LONG          lHandle  
);
```

参数

IHandle

[IN] 客户端句柄，由 [NET_ESS_CreateClient](#) 返回。

返回值

返回 *TRUE* 表示成功，返回 *FALSE* 表示失败。

如果返回 *FALSE*，请调用 [NET_ESS_GetLastError](#) 获取错误码。

第 15 章 回调函数

15.1 ASYNC_RESPONSE_CB

响应回调函数

回调函数定义

```
typedef BOOL (CALLBACK *ASYNC_RESPONSE_CB) (  
    NET_EHOME_ASYNC_RESP_CB_DATA lpData,  
    void *pUser  
);
```

参数

lpData

[OUT] 被回调的响应数据，详情参见结构体 [NET_EHOME_ASYNC_RESP_CB_DATA](#)。

pUser

[OUT] 用户参数指针。

相关接口

[NET_ECMS_SetXmlConfigResponseCB](#)

15.2 DEVICE_REGISTER_CB

注册回调函数。

回调函数定义

```
typedef BOOL (CALLBACK *DEVICE_REGISTER_CB) (  
    LONG lUserID,  
    DWORD dwDataType,  
    void *pOutBuffer,  
    DWORD dwOutLen,  
    void *pInBuffer,  
    DWORD dwInLen,  
    void *pUser  
);
```

参数

lUserID

[OUT] 用户 ID，由 SDK 分配。

dwDataType

[OUT] 回调数据类型，详见枚举 [NET_EHOME_REGISTER_TYPE](#)。对于支持不同版本 ISUP 的设备，其对应的回调数据类型逻辑不同，且每种数据类型需要以一定的顺序被回调，详见 [设备注册消息处理流程](#)。

pOutBuffer

[OUT] 输出缓冲区。不同的回调数据类型（dwDataType）对应的输出参数不同。

dwOutLen

[OUT] 输出缓冲区大小。

pInBuffer

[IN] 输入缓冲区，不同的输入参数对应不同的回调数据类型（dwDataType）。

dwInLen

[IN] 输入缓冲区大小。

pUser

[OUT] 用户参数。

备注

- 输入参数（pInBuffer）、回调数据类型（dwDataType）和输出参数（pOutBuffer）之间的关系如下所示。

dwDataType	pInBuffer	pOutBuffer
ENUM_DEV_ON	<u>NET_EHOME_SERVER_INFO</u>	<u>NET_EHOME_DEV_REG_INFO</u>
ENUM_DEV_OFF	无	无
ENUM_DEV_ADDRESS_CHANGED	NULL	<u>NET_EHOME_DEV_REG_INFO</u>
ENUM_EHOME50_DEV_AUTH	EHomeKey（调用 <u>NET_ECMS_GetDeviceSessionKey</u> 获取）	<u>NET_EHOME_DEV_REG_INFO</u>
ENUM_EHOME50_DEV_SESSIONKEY	无	<u>NET_EHOME_DEV_REG_INFO</u>

dwDataType	pInBuffer	pOutBuffer
ENUM_EHOME50_DEV_DAS_REQ	<u>JSON_DasInfo</u>	<u>NET_EHOME_DEV_REG_INFO</u>
ENUM_DEV_SESSIONKEY_REQ	<u>NET_EHOME_DEV_REG_INFO</u> <u>V12</u>	无
ENUM_DEV_DAS_REREGISTER	<u>NET_EHOME_DEV_REG_INFO</u> <u>V12</u>	<u>NET_EHOME_SERVER_INFO_V</u> <u>50</u>
ENUM_DEV_DAS_PINGREQ	<u>NET_EHOME_DEV_REG_INFO</u> <u>V12</u>	无
ENUM_DEV_DAS_EHOMEKEY_ERROR	无	无
ENUM_DEV_SESSIONKEY_ERROR	<u>NET_EHOME_DEV_REG_INFO</u> <u>V12</u>	无
ENUM_DEV_SLEEP	无	无

- 回调设备离线信息时，必须调用 NET_ECMS ForceLogout 注销登录。

15.3 ECMSCallback

中心管理服务器（CMS）的异步回调函数

回调函数定义

```
typedef void(CALLBACK *ECMSCallback) (
    NET_EHOME_CMSCB_DATA *pData,
    void *pUser
);
```

参数

pData

[OUT] 被回调的数据，详情参见结构体 NET_EHOME_CMSCB_DATA。

pUser

[OUT] 用户参数指针。

15.4 EHomeMsgCallBack

报警回调函数。

回调函数定义

```
typedef BOOL (CALLBACK *EHomeMsgCallBack) (  
    LONG                                iHandle,  
    NET_EHOME_ALARM_MSG                *pAlarmMsg,  
    void                                *pUser  
);
```

参数

iHandle

[OUT] 报警监听句柄。

pAlarmMsg

[OUT] 报警信息，详情参见结构体 [NET_EHOME_ALARM_MSG](#)。

pUser

[OUT] 用户参数。

相关接口

[NET_EALARM_StartListen](#)

15.5 EHomeSSMsgCallBack

文件信息回调函数

回调函数定义

```
typedef BOOL (CALLBACK *EHomeSSMsgCallBack) (  
    LONG                                iHandle,  
    NET_EHOME_SS_MSG_TYPE              enumType,  
    void                                *pOutBuffer,  
    DWORD                               dwOutLen,  
    void                                *pInBuffer,  
    DWORD                               dwInLen,  
    void                                *pUser  
);
```

参数

iHandle

[OUT] 句柄。

enumType

[IN] 回调数据类型，详见枚举定义 [NET_EHOME_SS_MSG_TYPE](#)。

pOutBuffer

[OUT] 保存输出参数的缓冲区，取决于被回调的数据的类型 (**enumType**)。

dwOutLen

[OUT] 输出缓冲区大小。

pInBuffer

[IN] 保存输入参数的缓冲区，取决于被回调的数据的类型 (**enumType**)。

dwInLen

[IN] 输入缓冲区大小。

pUser

[OUT] 用户参数。

相关接口

[NET_ECMS_StartListen](#)

备注

存储服务开启时，该回调函数 (EHomeSSMsgCallBack) 为必填。

15.6 EHomeSSRWCallBack

读写回调函数

回调函数定义

```
typedef BOOL(CALLBACK *EHomeSSRWCallBack) (  
    LONG            iHandle,  
    BYTE            byAct,  
    const char      *pFileName,  
    void            *pFileBuf,  
    LONG            *dwFileLen,  
    const char      *pFileUrl,  
    void            *pUser  
);
```

参数

iHandle

[OUT] 句柄。

byAct

[OUT] 操作类型：0-写，1-读，2-删除。

pFileName

[OUT] 文件名。

pFileBuf

[IN][OUT] 保存文件的缓冲区。

dwFileLen

[IN][OUT] 文件大小。

pFileUrl

[OUT] 文件 URL。

pUser

[OUT] 用户参数。

相关接口

[NET_ECMS_StartListen](#)

15.7 EHomeSSRWCallBackEx

读写回调函数扩展

回调函数定义

```
typedef BOOL (CALLBACK *EHomeSSRWCallBackEx) (  
    LONG                iHandle,  
    NET_EHOME_SS_RW_PARAM* pRwParam,  
    NET_EHOME_SS_EX_PARAM* pExStruct  
);
```

参数

iHandle

存储组件服务端监听句柄，[NET_ESS_StartListen](#) 的返回值。

pRwParam

用于传递四种存储协议共用字段，参见 [NET_EHOME_SS_RW_PARAM](#)。

pExStruct

用于传递存储协议特有字段，参见 [NET_EHOME_SS_EX_PARAM](#)。

15.8 EHomeSSStorageCallBack

存储信息回调函数

回调函数定义

```
typedef BOOL (CALLBACK *EHomeSSStorageCallBack) (  
    LONG                iHandle,  
    const char*         *pFileName,  
    void                *pFileBuf,  
    DWORD              dwFileLen,  
    char               *pFilePath,  
    void               *pUser  
);
```

参数

iHandle

[OUT] 存储句柄。

pFileName

[OUT] 文件名。

pFileBuf

[OUT] 保存文件的缓冲区大小。

pFilePath

[OUT] 文件保存路径。

dwInLen

[IN] 输入缓冲区大小。

pUser

[OUT] 用户参数。

相关接口

[NET_ECMS_StartListen](#)

15.9 fExceptionCallBack

接收异常信息的回调函数

回调函数定义

```
typedef void (CALLBACK *fExceptionCallBack) (
    DWORD    dwType,
    int      iUserID,
    int      iHandle,
    void     *pUser
);
```

参数

dwType

[OUT] 异常信息类型，详情如下：

宏定义	值	描述
EHOME_VOICETALK_EXCEPTION	0x101	语音对讲异常。
EHOME_PREVIEW_EXCEPTION	0x102	预览异常。
EHOME_PLAYBACK_EXCEPTION	0x103	回放异常。
EHOME_AUDIOTALK_EXCEPTION	0x104	音频取流异常。
EHOME_CMSALARM_EXCEPTION	0x105	报警接收异常。

lUserID

[OUT] 用户 ID。

iHandle

[OUT] 句柄（会话 ID）。

pUser

[OUT] 用户参数指针。

15.10 fVoiceDataCallBack

音频数据回调函数

接口定义

```
typedef void (CALLBACK *fVoiceDataCallBack) (
    LONG     iVoiceComHandle,
    char     *pRecvDataBuffer,
    int      dwBufSize,
```



```

int      dwEncodeType,
BYTE     byAudioFlag,
void     *pUser
);

```

参数

iVoiceComHandle

[OUT] 语音对讲或音频转发句柄，由 [NET_ECMS_StartVoiceTalk](#) 返回。

pRecvDataBuffer

[OUT] 保存音频文件的缓冲区的指针。

dwBufSize

[OUT] 音频数据大小。

dwEncodeType

[OUT] 音频数据编码类型：0-G.722，1-G.711U，2-G.711A，3-G.726，4-AAC，5-MP2L2，6-PCM。

byAudioFlag

[OUT] 音频数据源。0-本地采集，1-来自设备。

pUser

[OUT] 用户参数。

15.11 LongConfigCallback

长连接回调函数

接口定义

```

typedef BOOL(CALLBACK *LongConfigCallback)(
    LONG          iHandle,
    LONG_LINK_MSG enMsg,
    void          *pOutBuffer,
    DWORD         dwOutLen,
    void          *pUser
);

```

参数

iHandle

[OUT] 长连接句柄，由 [NET_ECMS_LongConfigCreate](#) 返回。

enMsg

[OUT] 长连接信息类型，枚举于 [LONG_LINK_MSG](#)。

pOutBuffer

[OUT] 输出缓冲区。

dwOutLen

[OUT] 输出缓冲区中的数据大小。

pUser

[OUT] 用户指针。

15.12 PASSTHROUGHDATAcallback

数据传输回调函数

接口定义

```
typedef void (CALLBACK* PASSTHROUGHDATAcallback) (  
    DWORD                dwProxyType,  
    LONG                 iListenHandle,  
    void                 pDeviceID,  
    DWORD                dwDevIDLen,  
    void                 pDataBuffer,  
    DWORD                dwDataLen,  
    void                 pUser  
);
```

参数

dwProxyType

[OUT] 代理类型，默认类型为“ENUM_PROXY_TYPE_NETSDK”（私有协议代理）。

iListenHandle

[OUT] 代理监听句柄，由 [NET_ECMS_StartListenProxy](#) 返回。

pDeviceID

[OUT] 设备 ID。

dwDevIDLen

[OUT] 设备 ID 长度。

pDataBuffer

[OUT] 输出缓冲区。

dwDataLen

[OUT] 输出缓冲区大小。

pUser

[OUT] 用户指针。

15.13 PLAYBACK_DATA_CB

回放数据的回调函数

接口定义

```
typedef BOOL (CALLBACK *PLAYBACK_DATA_CB) (  
    LONG                                iPlayBackLinkHandle,  
    NET_EHOME_PLAYBACK_DATA_CB_INFO    *pDataCBInfo,  
    void                                *pUserData  
);
```

参数

iPlayBackLinkHandle

[OUT] 回放句柄，由 [PLAYBACK_NEWLINK_CB](#) 返回。

pDataCBInfo

[OUT] 被回调的回放数据，详情参见 [NET_EHOME_PLAYBACK_DATA_CB_INFO](#)。

pUserData

[OUT] 用户参数。

返回值

返回 *TRUE* 表示开始该通道的回放，返回 *FALSE* 表示取消该通道的回放并自动释放资源。

15.14 PLAYBACK_NEWLINK_CB

回放请求的回调函数

回调函数定义

```
typedef BOOL (CALLBACK *PLAYBACK_NEWLINK_CB) (  
    LONG                                iPlayBackLinkHandle,  
    NET_EHOME_PLAYBACK_NEWLINK_CB_INFO *pNewLinkCBInfo,  
    void                                *pUserData  
);
```

参数

iPlayBackLinkHandle

[OUT] 回放句柄。

pNewLinkCBInfo

[OUT] 被回调的回放请求，详情参见结构体 [NET_EHOME_PLAYBACK_NEWLINK_CB_INFO](#)。

pUserData

[OUT] 用户参数。

返回值

返回 *TRUE* 表示开始该通道的回放，返回 *FALSE* 表示取消该通道的回放并自动释放资源。

相关接口

[NET_ESTREAM_StartListenPlayBack](#)

15.15 PREVIEW_DATA_CB

预览数据的回调函数

回调函数定义

```
typedef BOOL (CALLBACK *PREVIEW_DATA_CB) (  
    LONG                iPreviewHandle,  
    NET_EHOME_PREVIEW_CB_MSG *pPreviewCBMsg,  
    void                *pUserData  
);
```

参数

iPreviewHandle

[OUT] 预览句柄，由 [PREVIEW_NEWLINK_CB](#) 返回。

pPreviewCBMsg

[OUT] 被回调的预览数据，详情参见结构体 [NET_EHOME_PREVIEW_CB_MSG](#)

pUserData

[OUT] 用户参数。

返回值

返回 *TRUE* 表示开始该通道的预览，返回 *FALSE* 表示取消该通道的预览并自动释放资源。

15.16 PREVIEW_NEWLINK_CB

接收预览请求响应的回调函数

回调函数定义

```
typedef BOOL (CALLBACK *PREVIEW_NEWLINK_CB) (  
    LONG                                lLinkHandle,  
    NET_EHOME_NEWLINK_CB_MSG           *pNewLinkCBMsg,  
    void                                *pUserData  
);
```

参数

lLinkHandle

[OUT] 预览连接句柄。

pNewLinkCBMsg

[OUT] 预览请求参数，详情参见结构体 [NET_EHOME_NEWLINK_CB_MSG](#)。

pUserData

[OUT] 用户参数。

返回值

返回 *TRUE* 表示开始该通道的预览，返回 *FALSE* 表示取消该通道的预览并自动释放资源。

15.17 VOICETALK_DATA_CB

语音对讲数据的回调函数

接口定义

```
typedef BOOL (CALLBACK *VOICETALK_DATA_CB) (  
    LONG                                lHandle,  
    NET_EHOME_VOICETALK_DATA_CB_INFO   *pDataCBInfo,  
    void                                *pUserData  
);
```

参数

lHandle

[OUT] 语音对讲句柄，由 [VOICETALK_NEWLINK_CB](#) 返回。

pDataCBInfo

[OUT] 被回调的语音对讲数据，详情参见结构体 [NET_EHOME_VOICETALK_DATA_CB_INFO](#)。

pUserData

[OUT] 用户参数。

返回值

返回 *TRUE* 表示开始该通道的语音对讲，返回 *FALSE* 表示取消该通道的语音对讲并自动释放资源。

15.18 VOICETALK_NEWLINK_CB

语音对讲请求的回调函数

接口定义

```
typedef BOOL (CALLBACK *VOICETALK_NEWLINK_CB) (  
    LONG                                lHandle,  
    NET_EHOME_VOICETALK_NEWLINK_CB_INFO *pNewLinkCBInfo,  
    void                                 *pUser  
);
```

参数

lHandle

[OUT] 语音对讲句柄。

pNewLinkCBInfo

[OUT] 被回调的语音对讲参数，详情参见结构体 [NET_EHOME_VOICETALK_NEWLINK_CB_INFO](#)

。

pUser

[OUT] 用户参数。

返回值

返回 *TRUE* 表示开始该通道的语音对讲，返回 *FALSE* 表示取消该通道的语音对讲并自动释放资源。

第 16 章 命令和报文

16.1 控制命令报文

16.1.1 ADJUSTTIME

手动设备校时。

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>REQUEST</CommandType>
  <Method>CONTROL</Method>
  <Command>ADJUSTTIME</Command>
  <Params>
    <!--时区：0-与设备一致，1-GMT-12:00，2-GMT-11:00，…，15-GMT，16-GMT+01:00，17-GMT
+02:00，…，34-GMT+13:00-->
    <TimeZone>0</TimeZone>
    <!--时间，格式：YYYY-MM-DD HH:MM:SS-->
    <Time>2016-08-28 08:00:00</Time>
  </Params>
</PPVSPMessage>
```

响应报文

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <Method>CONTROL</Method>
  <WhichCommand>ADJUSTTIME</WhichCommand>
  <!--控制状态：200-成功，400-请求 XML 报文或通道号错误，500-失败，501-设备不支持手动校时-->
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.1.2 GETDEVICEWORKSTATUS

查找并获取设备工作状态。

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
  <PPVSPMessage>
    <Version>4.0</Version>
    <Sequence>1</Sequence>
    <CommandType>REQUEST</CommandType>
    <Method>QUERY</Method>
    <Command>GETDEVICEWORKSTATUS</Command>
    <Params><!--参数--></Params>
  </PPVSPMessage>
```

响应报文

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <Method>QUERY</Method>
  <WhichCommand>GETDEVICEWORKSTATUS</WhichCommand>
  <!--操作状态：200-成功，500-失败-->
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <DeviceStatusXML>
      <!--设备状态：0-正常，1-CPU使用率过高（高于85%），2-硬件错误-->
      <Run>0</Run>
      <!--CPU使用率，取值范围为0%到100%-->
      <CPU>45</CPU>
      <!--内存使用率，取值范围为0%到100%-->
      <Mem>98</Mem>
      <DSKStatus>
        <!--包含多个参数的字符串，包括硬盘号，硬盘容量（单位：MB），硬盘剩余空间（单位：MB），和硬盘状态（0-活跃，1-睡眠，2-异常）；每个参数由“-”分隔-->
        <DSK>1-60000-30000-0</DSK>
        <DSK>2-60000-30000-1</DSK>
        <DSK>2-32000-0-2</DSK>
      </DSKStatus>
      <CHStatus>
        <!--包含多个参数的字符串，包括通道号，录像状态（0-停止，1-开始），视频信号状态（0-正常，1-视频丢失），通道编码状态（0-正常，1-异常），实际码率（单位：Kbps），关联客户端数量，相机关联状态（0-未关联相机，1-已关联相机），和相机状态（0-离线，1-上线）；每个参数由“-”分隔-->
        <CH>1-0-0-0-256-1-1-1</CH>
        <CH>2-0-0-0-256-1-1-1</CH>
      </CHStatus>
      <!--已启用报警输入号，多个号之间用逗号隔开-->
      <AlarmInStatus>1,2,3,6,8</AlarmInStatus>
      <!--已启用报警输出号，多个号之间用逗号隔开-->
      <AlarmOutStatus>1,2,3,6,8</AlarmOutStatus>
```



```

<!--本地显示状态：0-正常，1-异常-->
<LocalDisplayStatus>0</LocalDisplayStatus>
<!--是否禁用预览：1-是，0-否（默认）-->
<ForbidPreview>0</ForbidPreview>
<!--全局布防状态：1-已布防，0-未布防（默认）-->
<DefenseStatus>1</DefenseStatus>
<!--布防延迟时间，单位：秒，默认值为0-->
<ArmDelayTime>0</ArmDelayTime>
<!--相机状态：0-异常，1-正常（默认），2-相机未使用，3-未知-->
<CameraSt>0</CameraSt>
<!--车底检测器状态：0-异常，1-正常（默认），2-检测器未使用，3-未知-->
<SensorsSt>1</SensorsSt>
<!--线圈状态：0-异常，1-正常（默认），2-线圈未使用，3-未知-->
<CoilSt>0</CoilSt>
<Remark><!--保留节点--></Remark>
</DeviceStatusXML>
</Params>
</PPVSPMessage>

```

16.1.3 PZIN

执行 3D 定位。

请求报文

```

<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>REQUEST</CommandType>
  <Method>CONTROL</Method>
  <Command>PZIN</Command>
  <Params>
    <!--需进行控制的通道号-->
    <Channel>1</Channel>
    <PTZCmd>PZIN</PTZCmd>
    <!--操作：0-缩小（从右上到左下，从右下到左上），1-放大（从左上到右下，从左下到右上）-->
    <Action>0</Action>
    <!--选择框起始点的 X 坐标-->
    <AreaX>150</AreaX>
    <!--选择框起始点的 Y 坐标-->
    <AreaY>114</AreaY>
    <!--选择框的宽度-->
    <AreaWidth>227</AreaWidth>
    <!--选择框的高度-->
    <AreaHeight>183</AreaHeight>
  </Params>
</PPVSPMessage>

```

响应报文

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <Method>CONTROL</Method>
  <WhichCommand>PZIN</WhichCommand>
  <!--控制状态：200-成功，400-请求 XML 报文或通道号错误，500-失败，501-设备不支持手动校时-->
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.1.4 REBOOT

重启设备。

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>REQUEST</CommandType>
  <Method>CONTROL</Method>
  <Command>REBOOT</Command>
</PPVSPMessage>
```

响应报文

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <Method>CONTROL</Method>
  <WhichCommand>REBOOT</WhichCommand>
  <!--控制状态：200-成功，500-失败-->
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.1.5 RESTARTIVSLIB

重启算法库。

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>REQUEST</CommandType>
  <Method>QUERY</Method>
  <Command>RESTARTIVSLIB</Command>
  <Params></Params>
</PPVSPMessage>
```

响应报文

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <Method>QUERY</Method>
  <WhichCommand>RESTARTIVSLIB</WhichCommand>
  <!--控制状态：200-成功，500-失败-->
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.1.6 SETDEFENSESTATUS

远程布防或撤防。

请求报文

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>REQUEST</CommandType>
  <Method>QUERY</Method>
  <Command>SETDEFENSESTATUS</Command>
  <Params>
    <IsEnable><!--是否布防或撤防：1-布防，0-撤防--></IsEnable>
    <ArmDelayTime>
      <!--布防延迟时间，单位：秒，其取值范围为0到300-->
    </ArmDelayTime>
  </Params>
</PPVSPMessage>
```

响应报文

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <Method>QUERY</Method>
  <WhichCommand>SETDEFENSESTATUS</WhichCommand>
  <!--控制状态：200-成功，500-失败-->
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.1.7 UPDATE

升级设备。

请求报文

```
<?xml version="1.0" encoding="GB2312" ?>
<PPVSPMessage>
  <Version>2.5</Version>
  <Sequence>1</Sequence>
  <CommandType>REQUEST</CommandType>
  <Command>UPDATE</Command>
  <Params>
    <!--FTP 升级服务器的 IP 地址-->
    <FTPServerIP>120.34.98.30</FTPServerIP>
    <!--FTP 升级服务器的端口号-->
    <FTPServerPort>21</FTPServerPort>
    <!--FTP 升级服务器的用户名-->
    <Account>user</Account>
    <!--FTP 升级服务器的密码-->
    <Password>12345</Password>
    <!--升级文件名-->
    <File>digicap.dav</File>
    <!--待升级设备关联的通道号-->
    <Channel>1</Channel>
  </Params>
</PPVSPMessage>
```

响应报文

```
<?xml version="1.0" encoding="GB2312" ?>
<PPVSPMessage>
  <Version>4</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
```

```
<WhichCommand>UPDATE</WhichCommand>
<!--响应状态码：200-处理请求成功，401-请求报文错误-->
<Status>200</Status>
<Description>OK</Description>
</PPVSPMessage>
```

16.2 配置命令报文

16.2.1 GetAlarmDeploymentTime

获取报警布防计划参数。

需调用的接口和命令（pCmdBuf）

接口：*NET_ECMS_XMLConfig*

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>GetAlarmDeploymentTime</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，1-视频丢失，2-视频遮挡，3-报警输入，4-报警输出，9-客流统计-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从1开始-->
  </ConfigParam2>
  <ConfigParam3>
    <!--星期：0-星期一，1-星期二，2-星期三，3-星期四，4-星期五，5-星期六，6-星期日-->
  </ConfigParam3>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <ALARMTIME><!--布防计划-->
        <TIME><!--布防计划中的时间段-->
          <StartTime><!--时间段的开始时间，格式：HH:MM（时:分）--></StartTime>
```

```

    <StopTime><!--时间段的结束时间，格式：HH:MM（时:分）--></StopTime>
  </TIME>
  <TIME>
    <StartTime>00:00</StartTime>
    <StopTime>00:00</StopTime>
  </TIME>
</ALARMTIME>
</ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.2 GetAlarmLinkageType

获取报警联动动作。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```

<Params>
  <ConfigCmd>GetAlarmLinkageType</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，1-视频丢失，2-视频遮挡，6-PIR报警，7-无线报警，8-紧急报警，9-
客流统计-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从1开始-->
  </ConfigParam2>
</Params>

```

输出参数（pOutBuf）

```

<?xml version="1.0" encoding="UTF-8"?>
  <PPVSPMessage>
    <Version>4.0</Version>
    <Sequence>1</Sequence>
    <CommandType>RESPONSE</CommandType>
    <WhichCommand>GETDEVICECONFIG</WhichCommand>
    <Status>200</Status>
    <Description>OK</Description>
    <Params>
      <ConfigXML>
        <LINKAGETYPE>
          <MonitorAlarm><!--是否启用报警上墙：0-否，1-是--></MonitorAlarm>
          <SoundAlarm><!--是否启用声音告警：0-否，1-是--></SoundAlarm>
          <Upload><!--是否上传至中心：0-否，1-是--></Upload>

```

```

<Alarmout><!--是否启用报警输出：0-否，1-是--></Alarmout>
<Email><!--是否启用邮件联动：0-否，1-是--></Email>
<ALARMOUT>
  <AnalogAlarmOut1><!--是否启用模拟报警输出口1：0-否，1-是--></
AnalogAlarmOut1>
  <AnalogAlarmOut2><!--是否启用模拟报警输出口2：0-否，1-是--></
AnalogAlarmOut2>
</ALARMOUT>
<AlarmName>
  <!--报警设备名称，该节点只有当 ConfigParam1 为 6，7 或 8 时有效-->
</AlarmName>
<IsUseAlarm>
  <!--是否启用该报警：0-否，1-是；该节点只有当 ConfigParam1 为 6 或 7 时有效-->
</IsUseAlarm>
<SoundLight>
  <!--是否触发无线警号：0-否，1-是；该节点只有当 ConfigParam1 为 6，7 或 8 时有效-->
</SoundLight>
</LINKAGETYPE>
</ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.3 GetAlarmTriggerCapture

获取抓拍联动动作参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>GetAlarmTriggerCapture</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，1-视频丢失，2-视频遮挡，3-报警输入，6-PIR 报警，7-无线报警，8-
紧急报警-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从 1 开始-->
  </ConfigParam2>
</Params>

```

输出参数 (pOutBuf)

```

<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>

```

```

<Sequence>1</Sequence>
<CommandType>RESPONSE</CommandType>
<WhichCommand>GETDEVICECONFIG</WhichCommand>
<Status>200</Status>
<Description>OK</Description>
<Params>
  <ConfigXML>
    <CAPTURECONFIG>
      <UseCapture><!--是否启用抓拍：0-否，1-是--></UseCapture>
      <SendTo><!--存储类型：1-只保存在中心，2-只保存在本地，3-保存在中心和本地--></
SendTo>
      <Interval><!--抓拍间隔，当前该节点无效--></Interval>
      <Resolution>
        <!--分辨率：0-自动，1-D1，2-CIF，3-QCIF，4-UXGA，5-SVGA，6-HD720P，7-VGA，8-
XVGA，9-HD900P，10-HD1080-->
      </Resolution>
      <Quality><!--图片质量：40-正常，60-较好，80-最好--></Quality>
      <Frequency><!--连续抓拍次数，取值范围为1到5--></Frequency>
      <CAPTURECHAN>
        <AnalogChan1><!--是否触发模拟通道1抓拍：0-否，1-是--></AnalogChan1>
        <DigitChan1><!--是否触发数字通道1抓拍：0-否，1-是--></DigitChan1>
      </CAPTURECHAN>
    </CAPTURECONFIG>
  </ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.4 GetAlarmTriggerRecord

获取录像联动动作参数。

需调用的接口和命令 (pCmdBuf)

接口：[NET_ECMS_XMLConfig](#)

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>GetAlarmTriggerRecord</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，6-PIR报警，7-无线报警，8-紧急报警-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从1开始-->
  </ConfigParam2>
</Params>

```


输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <RECORDCHAN>
        <AnalogChan1><!--是否触发模拟通道 1 录像：0-否，1-是--></AnalogChan1>
        <DigitChan1><!--是否触发数字通道 1 录像：0-否，1-是--></DigitChan1>
      </RECORDCHAN>
    </ConfigXML>
  </Params>
</PPVSPMessage>
```

16.2.5 GetCalibrationPara

获取客流统计摄像机的标定参数。

需调用的接口和命令 (pCmdBuf)

接口：[NET_ECMS_XMLConfig](#)

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>GetCalibrationPara</ConfigCmd>
  <ConfigParam1>
    <!--通道号，从 1 开始-->
  </ConfigParam1>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
```

```

<ConfigXML>
  <CALIBRATIONCFG>
    <Enable><!--是否启用标定--></Enable>
    <DemarcationLineList><!--标定线参数列表，当前只支持一条线-->
      <DemarcationLine>
        <!--标定线坐标；每条标定线由两个点确定，每个点的 x 坐标和 y 坐标的取值范围为 0 到
1000-->
      </DemarcationLine>
    </DemarcationLineList>
  </CALIBRATIONCFG>
</ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.6 GetDevAbility

获取设备能力。

需调用的接口和命令（pCmdBuf）

接口：[*NET_ECMS_XMLConfig*](#)

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```

<Params>
  <ConfigCmd>GetDevAbility</ConfigCmd>
  <ConfigParam1><!--能力类型：1-软件能力--></ConfigParam1>
  <ConfigParam2>0</ConfigParam2>
  <ConfigParam3>0</ConfigParam3>
  <ConfigParam4>0</ConfigParam4>
</Params>

```

输出参数（pOutBuf）

```

<Params>
  <ConfigXML>
    <SoftwareCapability>
      <PlayByTimeSupport><!--是否支持按时间回放：TRUE-是，FALSE-否--></
PlayByTimeSupport>
      <NewInviteStream>
        <!--是否支持 4.0 版本及以上 ISUP 的预览功能（发送预览请求后传输实时码流）：TRUE-是，
FALSE-否-->
        <HRUDP><!--是否支持可靠码流传输：TRUE-是，FALSE-否--></HRUDP>
      </NewInviteStream>
      <P2PLinkNum>
        <!--支持的 P2P 连接数量，如果该节点不存在或值为 0，那么说明不支持 P2P 连接-->
      </P2PLinkNum >
      <VoiceTalkWithStmServer><!--是否支持通过流媒体服务器（SMS）开启语音对讲：TRUE-是，

```

```
FALSE-否--></VoiceTalkWithStmServer>
    <NewPlayBack>
        <!--是否支持 4.0 版本及以上 ISUP 的回放功能（发送回放请求后传输回放码流）：TRUE-是，
FALSE-否-->
    </NewPlayBack>
</SoftwareCapability>
</ConfigXML>
</Params>
```

16.2.7 GetExceptionPara

获取异常报警联动参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
    <ConfigCmd>GetExceptionPara</ConfigCmd>
    <!--异常类型：0-硬盘满，1-硬盘错误，2-网络断开，3-IP 地址冲突，4-非法登录，5-输入和输出视频
标准不匹配，6-视频信号异常，7-视频异常-->
    <ConfigParam1></ConfigParam1>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
    <Version>4.0</Version>
    <Sequence>1</Sequence>
    <CommandType>RESPONSE</CommandType>
    <WhichCommand>GETDEVICECONFIG</WhichCommand>
    <Status>200</Status>
    <Description>OK</Description>
    <Params>
        <ConfigXML>
            <EXCEPTIONINFO>
                <LINKAGETYPE>
                    <MonitorAlarm><!--是否启用报警上墙：0-否，1-是--></MonitorAlarm>
                    <SoundAlarm><!--是否启用声音告警：0-否，1-是--></SoundAlarm>
                    <Upload><!--是否上传至中心：0-否，1-是--></Upload>
                    <Alarmout><!--是否启用报警输出：0-否，1-是--></Alarmout>
                    <Email><!--是否启用邮件联动：0-否，1-是--></Email>
                </ALARMOUT>
                <AnalogAlarmOut1><!--是否启用模拟报警输出口 1：0-否，1-是--></
```

```
AnalogAlarmOut1>
    <AnalogAlarmOut2><!--是否启用模拟报警输出口 2 : 0-否, 1-是--></
AnalogAlarmOut2>
    </ALARMOUT>
    </LINKAGETYPE>
    </EXCEPTIONINFO>
</ConfigXML>
</Params>
</PPVSPMessage>
```

16.2.8 GetHideAlarmArea

获取视频遮挡侦测的区域参数。

需调用的接口和命令（pCmdBuf）

接口：[*NET_ECMS_XMLConfig*](#)

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
    <ConfigCmd>GetHideAlarmArea</ConfigCmd>
    <!--通道号, 从 1 开始-->
    <ConfigParam1></ConfigParam1>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
    <Version>4.0</Version>
    <Sequence>1</Sequence>
    <CommandType>RESPONSE</CommandType>
    <WhichCommand>GETDEVICECONFIG</WhichCommand>
    <Status>200</Status>
    <Description>OK</Description>
    <Params>
        <ConfigXML>
            <HIDEALARMAREACFG>
                <AREARECT>
                    <AREALEFT><!--检测区域左上角顶点的 X 坐标--></AREALEFT>
                    <AREATOP><!--检测区域左上角顶点的 Y 坐标--></AREATOP>
                    <AREARIGHT><!--检测区域右下角顶点的 X 坐标--></AREARIGHT>
                    <AREABOTTOM><!--检测区域右下角顶点的 Y 坐标--></AREABOTTOM>
                </AREARECT>
            </HIDEALARMAREACFG>
        </ConfigXML>
    </Params>
</PPVSPMessage>
```

```
</Params>  
</PPVSPMessage>
```

16.2.9 GetHideAlarmPara

获取视频遮挡侦测参数。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```
<Params>  
  <ConfigCmd>GetHideAlarmPara</ConfigCmd>  
  <!--通道号，从1开始-->  
  <ConfigParam1></ConfigParam1>  
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTF-8"?>  
<PPVSPMessage>  
  <Version>4.0</Version>  
  <Sequence>1</Sequence>  
  <CommandType>RESPONSE</CommandType>  
  <WhichCommand>GETDEVICECONFIG</WhichCommand>  
  <Status>200</Status>  
  <Description>OK</Description>  
  <Params>  
    <ConfigXML>  
      <HIDEALARMCFG>  
        <Sensitive><!--视频遮挡侦测的灵敏度等级：0-禁用，1-低，2-中，3-高--></Sensitive>  
      </HIDEALARMCFG>  
    </ConfigXML>  
  </Params>  
</PPVSPMessage>
```

16.2.10 GetHideArea

获取隐私遮蔽区域参数。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>GetHideArea</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <HIDEAREACFG>
        <AREARECT>
          <AREALEFT><!--检测区域左上角顶点的 x 坐标--></AREALEFT>
          <AREATOP><!--检测区域左上角顶点的 y 坐标--></AREATOP>
          <AREARIGHT><!--检测区域右下角顶点的 x 坐标--></AREARIGHT>
          <AREABOTTOM><!--检测区域右下角顶点的 y 坐标--></AREABOTTOM>
        </AREARECT>
        <AREARECT>
          <AREALEFT></AREALEFT>
          <AREATOP></AREATOP>
          <AREARIGHT></AREARIGHT>
          <AREABOTTOM></AREABOTTOM>
        </AREARECT>
        <AREARECT>
          <AREALEFT></AREALEFT>
          <AREATOP></AREATOP>
          <AREARIGHT></AREARIGHT>
          <AREABOTTOM></AREABOTTOM>
        </AREARECT>
        <AREARECT>
          <AREALEFT></AREALEFT>
          <AREATOP></AREATOP>
          <AREARIGHT></AREARIGHT>
          <AREABOTTOM></AREABOTTOM>
        </AREARECT>
      </HIDEAREACFG>
    </ConfigXML>
  </Params>
</PPVSPMessage>
```

16.2.11 GetHidePara

获取隐私遮蔽参数。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>GetHidePara</ConfigCmd>
  <!--通道号, 从1开始-->
  <ConfigParam1></ConfigParam1>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <HIDECFG>
        <IsUseHide><!--是否启用视频隐私遮蔽：0-否，1-是--></IsUseHide>
      </HIDECFG>
    </ConfigXML>
  </Params>
</PPVSPMessage>
```

16.2.12 GetMotionArea

获取移动侦测区域参数。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：GETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>GetMotionArea</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <MOTIONAREACFG>
        <Row><!--区域中的行数--></Row>
        <BlockPerRow><!--区域中的列数--></BlockPerRow>
        <PictureWidth><!--图片宽度, 当前该节点将被设置为 0--></PictureWidth>
        <PictureHeight><!--图片高度, 当前该节点将被设置为 0--></PictureHeight>
        <AREAS>
          <!--<Mask>节点数应与行数相同, <Mask>节点的取值按位表示 (位数与列数相同); 每一位的
          取值为 1 或 0, 表示该网格启用或禁用移动侦测-->
          <Mask></Mask>
          <Mask></Mask>
          <Mask></Mask>
          <Mask></Mask>
        </AREAS>
      </MOTIONAREACFG>
    </ConfigXML>
  </Params>
</PPVSPMessage>
```

16.2.13 GetMotionPara

获取移动侦测参数。

需调用的接口和命令 (pCmdBuf)

接口 : NET_ECMS_XMLConfig

命令 (pCmdBuf) : GETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>GetMotionPara</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <MOTIONCFG>
        <!--是否启用移动侦测: 0-否, 1-是-->
        <IsUseMotion></IsUseMotion>
        <!--灵敏度等级: 255-禁用, 0-1级, 1-2级, 2-3级, 3-4级, 4-5级, 5-6级-->
        <Sensitive></Sensitive>
      </MOTIONCFG>
    </ConfigXML>
  </Params>
</PPVSPMessage>
```

16.2.14 GetPassengerLinePara

获取客流统计检测线参数。

需调用的接口和命令（pCmdBuf）

接口：[*NET_ECMS_XMLConfig*](#)

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>GetPassengerLinePara</ConfigCmd>
  <ConfigParam1>
    <!--通道号, 从 1 开始-->
  </ConfigParam1>
  <ConfigParam2>
    <!--检测线号, 从 1 开始; 但当前只支持一条检测线-->
  </ConfigParam2>
</Params>
```

```
</ConfigParam2>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <PASSENGERLINECFG>
        <Sensitivity><!--目标检测灵敏度, 取值范围: [1, 100]--></Sensitivity>
        <SpaceSpeed><!--目标生成速度 (空间), 取值范围: [1, 100]--></SpaceSpeed>
        <TimeSpeed><!--目标生成速度 (时间), 取值范围: [1, 100]--></TimeSpeed>
        <CountSpeed><!--统计速度, 取值范围: [1, 100]--></CountSpeed>
        <DetectType><!--待检测的目标类型: 0-自动, 1-头部, 2-头部和肩部--></DetectType>
        <SizeCorrect><!--检测线长度的校正因子, 取值范围为 1 到 100-->
        </SizeCorrect>
        <LineCoordinatesList>
          <!--检测线坐标; 每条检测线由两个点确定, 每个点的 x 坐标和 y 坐标的取值范围为 0 到 1000-->
        </LineCoordinatesList>
        <Direction>
          <!--进入方向, 由两个点确定, 每个点的 x 坐标和 y 坐标的取值范围为 0 到 1000-->
        </Direction>
      </PASSENGERLINECFG>
    </ConfigXML>
  </Params>
</PPVSPMessage>
```

16.2.15 GetPassengerOSDPara

获取客流统计信息的 OSD (屏幕菜单式调节方式) 参数。

需调用的接口和命令 (pCmdBuf)

接口: NET_ECMS_XMLConfig

命令 (pCmdBuf): GETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>GetPassengerOSDPara</ConfigCmd>
  <ConfigParam1>
```

```

    <!--通道号, 从1开始-->
  </ConfigParam1>
</Params>

```

输出参数 (pOutBuf)

```

<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <PASSENGEROSDCFG>
        <Enable><!--是否启用客流统计信息 OSD: 0-否, 1-是--></Enable>
        <OSDPos><!--显示位置左上角的坐标, x 坐标和 y 坐标的取值范围为 0 到 1000--></OSDPos>
      </PASSENGEROSDCFG>
    </ConfigXML>
  </Params>
</PPVSPMessage>

```

16.2.16 GetPassengerPara

获取客流统计参数。

需调用的接口和命令 (pCmdBuf)

接口：[*NET_ECMS_XMLConfig*](#)

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>GetPassengerPara</ConfigCmd>
  <ConfigParam1>
    <!--通道号, 从1开始-->
  </ConfigParam1>
</Params>

```

输出参数 (pOutBuf)

```

<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>

```

```

<WhichCommand>GETDEVICECONFIG</WhichCommand>
<Status>200</Status>
<Description>OK</Description>
<Params>
  <ConfigXML>
    <PASSENGERCFG>
      <Enable><!--是否启用客流统计：0-否，1-0是--></Enable>
    </PASSENGERCFG>
  </ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.17 GetRecordPlanPara

获取录像参数。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：GETDEVICECONFIG

输入参数（pInBuf）

```

<Params>
  <ConfigCmd>GetRecordPlanPara</ConfigCmd>
  <ConfigParam1>
    <!--通道号，从1开始-->
  </ConfigParam1>
  <ConfigParam2>
    <!--星期：0-星期一，1-星期二，2-星期三，3-星期四，4-星期五，5-星期六，6-星期日-->
  </ConfigParam2>
</Params>

```

输出参数（pOutBuf）

```

<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <RECORDTIMECFG>
        <IsUseRecord><!--是否启用录像：0-否，1-是--></IsUseRecord>
        <RecordTime><!--延录时长：0-5s，1-10s，2-30s，3-60s，4-120s，5-300s，6-600s-->
      </RecordTime>
    </ConfigXML>
  </Params>
</PPVSPMessage>

```

```

<PreRecordTime>
  <!--预录时长：0-不预录，1-5s，2-10s，3-15s，4-20s，5-25s，6-30s，7-无限制-->
</PreRecordTime>
<RecorderDuration>
  <!--录像视频的最大保存时间（只读），该时间的取值范围为1到60天，0表示长期保存-->
</RecorderDuration>
<RedundancyRec><!--是否为冗余视频（只读）：0-否，1-是--></RedundancyRec>
<AudioRec><!--是否录制音频数据（只读）：0-否，1-是--></AudioRec>
<RECORDPLAY>
  <IsAllDayRecord><!--是否处于全天录像模式：0-否，1-是--></IsAllDayRecord>
  <AllDayRecordType>
    <!--录像触发类型：0-计划录像，1-移动侦测，2-报警，3-移动侦测或报警，4-移动侦测和报警，5-命令，6-手动，7-震动报警，8-场景变更报警，9-VCA报警，10-PIR报警，11-无线报警，12-紧急报警，12-所有报警-->
  </AllDayRecordType>
  <TIME><!--录像计划中的时间段-->
    <RecordType><!--录像触发类型，与AllDayRecordType相同--></RecordType>
    <StartTime><!--开始时间，例如：12:00--></StartTime>
    <StopTime><!--结束时间，例如：12:00--></StopTime>
  </TIME>
  <TIME>
    <RecordType></RecordType>
    <StartTime></StartTime>
    <StopTime></StopTime>
  </TIME>
</RECORDPLAY>
</RECORDTIMECFG>
</ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.18 GetRS232Para

获取 RS-232 串口参数。

需调用的接口和命令 (pCmdBuf)

接口：[NET_ECMS_XMLConfig](#)

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>GetRS232Para</ConfigCmd>
</Params>

```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
  <Params>
    <ConfigXML>
      <RS232INFO>
        <BaudRate>
          <!--波特率：0-50, 1-75, 2-110, 3-150, 4-300, 5-600, 6-1200, 7-2400, 8-4800,
          9-9600, 10-19200, 11-38400, 12-57600, 13-76800, 14-115.2K-->
        </BaudRate>
        <DataBit><!--数据位个数：0-5 位, 1-6 位, 2-7 位, 3-8 位--></DataBit>
        <StopBit><!--停止位个数：0-1 位, 1-2 位--></StopBit>
        <Parity><!--校验类型：0-无, 1-奇校验, 2-偶校验--></Parity>
        <FlowControl><!--流控制类型：0-无, 1-软件流控制, 2-硬件流控制--></FlowControl>
        <WorkMode><!--工作模式：0-用于 PPP 拨号, 1-用于参数控制, 2-作为透明通道--></
WorkMode>
      </RS232INFO>
    </ConfigXML>
  </Params>
</PPVSPMessage>
```

16.2.19 GetRS485Para

获取 RS-485 串口参数。

需调用的接口和命令 (pCmdBuf)

接口：[*NET_ECMS_XMLConfig*](#)

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>GetRS485Para</ConfigCmd>
  <ConfigParam1><!--通道号, 从 1 开始--></ConfigParam1>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
```

```

<Version>4.0</Version>
<Sequence>1</Sequence>
<CommandType>RESPONSE</CommandType>
<WhichCommand>GETDEVICECONFIG</WhichCommand>
<Status>200</Status>
<Description>OK</Description>
<Params>
  <ConfigXML>
    <RS485INFO>
      <BaudRate>
        <!--波特率：0-50, 1-75, 2-110, 3-150, 4-300, 5-600, 6-1200, 7-2400, 8-4800,
9-9600, 10-19200, 11-38400, 12-57600, 13-76800, 14-115.2K-->
      </BaudRate>
      <DataBit><!--数据位个数：0-5位, 1-6位, 2-7位, 3-8位--></DataBit>
      <StopBit><!--停止位个数：0-1位, 1-2位--></StopBit>
      <Parity><!--校验类型：0-无, 1-奇校验, 2-偶校验--></Parity>
      <FlowControl><!--流控制类型：0-无, 1-软件流控制, 2-硬件流控制--></FlowControl>
      <DecoderType><!--解码器协议类型--></DecoderType>
      <DecoderAddr><!--解码器地址, 取值范围为0到255--></DecoderAddr>
    </RS485INFO>
  </ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.20 GetVILostPara

获取视频丢失侦测参数。

需调用的接口和命令 (pCmdBuf)

接口：[NET_ECMS_XMLConfig](#)

命令 (pCmdBuf)：GETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>GetVILostPara</ConfigCmd>
  <!--通道号, 从1开始-->
  <ConfigParam1></ConfigParam1>
</Params>

```

输出参数 (pOutBuf)

```

<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>GETDEVICECONFIG</WhichCommand>

```

```

<Status>200</Status>
<Description>OK</Description>
<Params>
  <ConfigXML>
    <VILOSTCFG>
      <IsUseVILost><!--是否启用视频丢失侦测：0-否，1-是--></IsUseVILost>
    </VILOSTCFG>
  </ConfigXML>
</Params>
</PPVSPMessage>

```

16.2.21 SetAlarmDeploymentTime

设置报警布防计划参数。

需调用的接口和命令 (pCmdBuf)

接口：[*NET_ECMS_XMLConfig*](#)

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>SetAlarmDeploymentTime</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，1-视频丢失，2-视频遮挡，3-报警输入，4-报警输出，9-客流统计-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从1开始-->
  </ConfigParam2>
  <ConfigParam3>
    <!--星期：0-星期一，1-星期二，2-星期三，3-星期四，4-星期五，5-星期六，6-星期日-->
  </ConfigParam3>
  <ConfigXML>
    <ALARMTIME><!--布防计划-->
      <TIME><!--布防计划中的时间段-->
        <StartTime><!--时间段的开始时间，格式：HH:MM (时:分) --></StartTime>
        <StopTime><!--时间段的结束时间，格式：HH:MM (时:分) --></StopTime>
      </TIME>
      <TIME>
        <StartTime>00:00</StartTime>
        <StopTime>00:00</StopTime>
      </TIME>
    </ALARMTIME>
  </ConfigXML>
</Params>

```


输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.22 SetAlarmLinkageType

设置报警联动动作。

需调用的接口和命令 (pCmdBuf)

接口：[NET_ECMS_XMLConfig](#)

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>SetAlarmLinkageType</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，1-视频丢失，2-视频遮挡，6-PIR报警，7-无线报警，8-紧急报警，9-
    客流统计-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从1开始-->
  </ConfigParam2>
  <ConfigXML>
    <LINKAGETYPE>
      <MonitorAlarm><!--是否启用报警上墙：0-否，1-是--></MonitorAlarm>
      <SoundAlarm><!--是否启用声音告警：0-否，1-是--></SoundAlarm>
      <Upload><!--是否上传至中心：0-否，1-是--></Upload>
      <Alarmout><!--是否启用报警输出：0-否，1-是--></Alarmout>
      <Email><!--是否启用邮件联动：0-否，1-是--></Email>
      <ALARMOUT>
        <AnalogAlarmOut1><!--是否启用模拟报警输出口1：0-否，1-是--></AnalogAlarmOut1>
        <AnalogAlarmOut2><!--是否启用模拟报警输出口2：0-否，1-是--></AnalogAlarmOut2>
      </ALARMOUT>
      <AlarmName>
        <!--报警设备名称，该节点只有当 ConfigParam1 为 6，7 或 8 时有效-->
      </AlarmName>
      <IsUseAlarm>
        <!--是否启用该报警：0-否，1-是；该节点只有当 ConfigParam1 为 6 或 7 时有效-->
      </IsUseAlarm>
    </LINKAGETYPE>
  </ConfigXML>
</Params>
```

```

    </IsUseAlarm>
    <SoundLight>
      <!--是否触发无线警号：0-否，1-是；该节点只有当 ConfigParam1 为 6，7 或 8 时有效-->
    </SoundLight>
  </LINKAGETYPE>
</ConfigXML>
</Params>

```

输出参数（pOutBuf）

```

<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>

```

16.2.23 SetAlarmTriggerCapture

设置抓拍联动动作参数。

需调用的接口和命令（pCmdBuf）

接口：[NET_ECMS_XMLConfig](#)

命令（pCmdBuf）：SETDEVICECONFIG

输入参数（pInBuf）

```

<Params>
  <ConfigCmd>SetAlarmTriggerCapture</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，1-视频丢失，2-视频遮挡，3-报警输入，6-PIR 报警，7-无线报警，8-
    紧急报警-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从 1 开始-->
  </ConfigParam2>
  <ConfigXML>
    <CAPTURECONFIG>
      <UseCapture><!--是否启用抓拍：0-否，1-是--></UseCapture>
      <SendTo><!--存储类型：1-只保存在中心，2-只保存在本地，3-保存在中心和本地--></SendTo>
      <Interval><!--抓拍间隔，当前该节点无效--></Interval>
      <Resolution>
        <!--分辨率：0-自动，1-D1，2-CIF，3-QCIF，4-UXGA，5-SVGA，6-HD720P，7-VGA，8-
        XVGA，9-HD900P，10-HD1080-->
      </Resolution>
    </CAPTURECONFIG>
  </ConfigXML>
</Params>

```

```
<Quality><!--图片质量：40-正常，60-较好，80-最好--></Quality>
<Frequency><!--连续抓拍次数，取值范围为1到5--></Frequency>
<CAPTURECHAN>
  <AnalogChan1><!--是否触发模拟通道1抓拍：0-否，1-是--></AnalogChan1>
  <DigitChan1><!--是否触发数字通道1抓拍：0-否，1-是--></DigitChan1>
</CAPTURECHAN>
</CAPTURECONFIG>
</ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.24 SetAlarmTriggerRecord

设置录像联动动作参数。

需调用的接口和命令 (pCmdBuf)

接口：[*NET_ECMS_XMLConfig*](#)

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>SetAlarmTriggerRecord</ConfigCmd>
  <ConfigParam1>
    <!--报警类型：0-移动侦测，6-PIR报警，7-无线报警，8-紧急报警-->
  </ConfigParam1>
  <ConfigParam2>
    <!--通道号，从1开始-->
  </ConfigParam2>
  <ConfigXML>
    <RECORDCHAN>
      <AnalogChan1><!--是否触发模拟通道1录像：0-否，1-是--></AnalogChan1>
      <DigitChan1><!--是否触发数字通道1录像：0-否，1-是--></DigitChan1>
    </RECORDCHAN>
  </ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.25 SetCalibrationPara

设置客流统计摄像机的标定参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>SetCalibrationPara</ConfigCmd>
  <ConfigParam1>
    <!--通道号, 从 1 开始-->
  </ConfigParam1>
  <ConfigXML>
    <CALIBRATIONCFG>
      <Enable><!--是否启用标定--></Enable>
      <DemarcationLineList><!--标定线参数列表, 当前只支持一条线-->
        <DemarcationLine>
          <!--标定线坐标; 每条标定线由两个点确定, 每个点的 x 坐标和 y 坐标的取值范围为 0 到 1000-->
        </DemarcationLine>
      </DemarcationLineList>
    </CALIBRATIONCFG>
  </ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
```

```
<WhichCommand>SETDEVICECONFIG</WhichCommand>
<Status>200</Status>
<Description>OK</Description>
</PPVSPMessage>
```

16.2.26 SetExceptionPara

设置异常报警联动参数。

需调用的接口和命令 (pCmdBuf)

接口：[NET_ECMS_XMLConfig](#)

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>SetExceptionPara</ConfigCmd>
  <!--异常类型：0-硬盘满，1-硬盘错误，2-网络断开，3-IP地址冲突，4-非法登录，5-输入和输出视频
  标准不匹配，6-视频信号异常，7-视频异常-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <EXCEPTIONINFO>
      <LINKAGETYPE>
        <MonitorAlarm><!--是否启用报警上墙：0-否，1-是--></MonitorAlarm>
        <SoundAlarm><!--是否启用声音告警：0-否，1-是--></SoundAlarm>
        <Upload><!--是否上传至中心：0-否，1-是--></Upload>
        <Alarmout><!--是否启用报警输出：0-否，1-是--></Alarmout>
        <Email><!--是否启用邮件联动：0-否，1-是--></Email>
      <ALARMOUT>
        <AnalogAlarmOut1><!--是否启用模拟报警输出口1--></AnalogAlarmOut1>
        <AnalogAlarmOut2><!--是否启用模拟报警输出口2--></AnalogAlarmOut2>
      </ALARMOUT>
    </LINKAGETYPE>
  </EXCEPTIONINFO>
</ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
```

```
<Description>OK</Description>
</PPVSPMessage>
```

16.2.27 SetHideAlarmArea

设置视频遮挡侦测的区域参数。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：SETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>SetHideAlarmArea</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <HIDEALARMAREACFG>
      <AREARECT>
        <AREALEFT><!--检测区域左上角顶点的 x 坐标--></AREALEFT>
        <AREATOP><!--检测区域左上角顶点的 y 坐标--></AREATOP>
        <AREARIGHT><!--检测区域右下角顶点的 x 坐标--></AREARIGHT>
        <AREABOTTOM><!--检测区域右下角顶点的 y 坐标--></AREABOTTOM>
      </AREARECT>
    </HIDEALARMAREACFG>
  </ConfigXML>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.28 SetHideAlarmPara

设置视频遮挡侦测参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>SetHideAlarmPara</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <HIDEALARMCFG>
      <Sensitive><!--视频遮挡侦测的灵敏度等级: 0-禁用, 1-低, 2-中, 3-高--></Sensitive>
    </HIDEALARMCFG>
  </ConfigXML>
</Params>

```

输出参数 (pOutBuf)

```

<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>

```

16.2.29 SetHideArea

设置隐私遮蔽区域参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>SetHideArea</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <HIDEAREACFG>
      <AREARECT>

```

```
<AREALEFT><!--检测区域左上角顶点的 X 坐标--></AREALEFT>
<AREATOP><!--检测区域左上角顶点的 Y 坐标--></AREATOP>
<AREARIGHT><!--检测区域右下角顶点的 X 坐标--></AREARIGHT>
<AREABOTTOM><!--检测区域右下角顶点的 Y 坐标--></AREABOTTOM>
</AREARECT>
<AREARECT>
  <AREALEFT></AREALEFT>
  <AREATOP></AREATOP>
  <AREARIGHT></AREARIGHT>
  <AREABOTTOM></AREABOTTOM>
</AREARECT>
<AREARECT>
  <AREALEFT></AREALEFT>
  <AREATOP></AREATOP>
  <AREARIGHT></AREARIGHT>
  <AREABOTTOM></AREABOTTOM>
</AREARECT>
<AREARECT>
  <AREALEFT></AREALEFT>
  <AREATOP></AREATOP>
  <AREARIGHT></AREARIGHT>
  <AREABOTTOM></AREABOTTOM>
</AREARECT>
</HIDEAREACFG>
</ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.30 SetHidePara

设置隐私遮蔽参数。

需调用的接口和命令 (pCmdBuf)

接口：[*NET_ECMS_XMLConfig*](#)

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>SetHidePara</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <HIDECFG>
      <IsUseHide><!--是否启用视频隐私遮蔽: 0-否, 1-是--></IsUseHide>
    </HIDECFG>
  </ConfigXML>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.31 SetMotionArea

设置移动侦测区域参数。

需调用的接口和命令（pCmdBuf）

接口：[NET_ECMS_XMLConfig](#)

命令（pCmdBuf）：SETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>SetMotionArea</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <MOTIONAREACFG>
      <MOTIONAREACFG>
        <Row><!--区域中的行数--></Row>
        <BlockPerRow><!--区域中的列数--></BlockPerRow>
        <PictureWidth><!--图片宽度, 当前该节点将被设置为 0--></PictureWidth>
        <PictureHeight><!--图片高度, 当前该节点将被设置为 0--></PictureHeight>
      </MOTIONAREACFG>
    </MOTIONAREACFG>
  </ConfigXML>
</Params>
```

```

    <!--<Mask>节点数应与行数相同，<Mask>节点的取值按位表示（位数与列数相同）；每一位的取
    值为1或0，表示该网格启用或禁用移动侦测-->
    <Mask></Mask>
    <Mask></Mask>
    <Mask></Mask>
    <Mask></Mask>
  </AREAS>
</MOTIONAREACFG>
</ConfigXML>
</Params>

```

输出参数（pOutBuf）

```

<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>`
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>

```

16.2.32 SetMotionPara

设置移动侦测参数。

需调用的接口和命令（pCmdBuf）

接口：*NET_ECMS_XMLConfig*

命令（pCmdBuf）：SETDEVICECONFIG

输入参数（pInBuf）

```

<Params>
  <ConfigCmd>SetMotionPara</ConfigCmd>
  <!--通道号，从1开始-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <MOTIONCFG>
      <!--是否启用移动侦测：0-否，1-是-->
      <IsUseMotion></IsUseMotion>
      <!--灵敏度等级：255-禁用，0-1级，1-2级，2-3级，3-4级，4-5级，5-6级-->
      <Sensitive></Sensitive>
    </MOTIONCFG>
  </ConfigXML>
</Params>

```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>`
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.33 SetPassengerLinePara

设置客流统计检测线参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>SetPassengerLinePara</ConfigCmd>
  <ConfigParam1>
    <!--通道号, 从 1 开始-->
  </ConfigParam1>
  <ConfigParam2>
    <!--检测线号, 从 1 开始; 但当前只支持一条检测线-->
  </ConfigParam2>
  <ConfigXML>
    <PASSENGERLINECFG>
      <Sensitivity><!--目标检测灵敏度, 取值范围: [1, 100]--></Sensitivity>
      <SpaceSpeed><!--目标生成速度 (空间), 取值范围: [1, 100]--></SpaceSpeed>
      <TimeSpeed><!--目标生成速度 (时间), 取值范围: [1, 100]--></TimeSpeed>
      <CountSpeed><!--统计速度, 取值范围: [1, 100]--></CountSpeed>
      <DetectType><!--待检测的目标类型: 0-自动, 1-头部, 2-头部和肩部--></DetectType>
      <SizeCorrect><!--检测线长度的校正因子, 取值范围为 1 到 100-->
      </SizeCorrect>
      <LineCoordinatesList>
        <!--检测线坐标; 每条检测线由两个点确定, 每个点的 x 坐标和 y 坐标的取值范围为 0 到 1000-->
      </LineCoordinatesList>
      <Direction>
        <!--进入方向, 由两个点确定, 每个点的 x 坐标和 y 坐标的取值范围为 0 到 1000-->
      </Direction>
    </PASSENGERLINECFG>
```

```
</ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.34 SetPassengerOSDPara

设置客流统计信息的 OSD (屏幕菜单式调节方式) 参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>SetPassengerOSDPara</ConfigCmd>
  <ConfigParam1>
    <!--通道号, 从 1 开始-->
  </ConfigParam1>
  <ConfigXML>
    <PASSENGEROSDCFG>
      <Enable><!--是否启用客流统计信息 OSD: 0-否, 1-是--></Enable>
      <OSDPos><!--显示位置左上角的坐标, x 坐标和 y 坐标的取值范围为 0 到 1000--></OSDPos>
    </PASSENGEROSDCFG>
  </ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
```

```
<Description>OK</Description>
</PPVSPMessage>
```

16.2.35 SetPassengerPara

设置客流统计参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```
<Params>
  <ConfigCmd>SetPassengerPara</ConfigCmd>
  <ConfigParam1>
    <!--通道号, 从 1 开始-->
  </ConfigParam1>
  <ConfigXML>
    <PASSENGERCFG>
      <Enable><!--是否启用客流统计: 0-否, 1-0 是--></Enable>
    </PASSENGERCFG>
  </ConfigXML>
</Params>
```

输出参数 (pOutBuf)

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.36 SetRecordPlanPara

设置录像参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数（pInBuf）

```

<Params>
  <ConfigCmd>SetRecordPlanPara</ConfigCmd>
  <ConfigParam1>
    <!--通道号，从1开始-->
  </ConfigParam1>
  <ConfigParam2>
    <!--星期：0-星期一，1-星期二，2-星期三，3-星期四，4-星期五，5-星期六，6-星期日-->
  </ConfigParam2>
  <ConfigXML>
    <RECORDTIMECFG>
      <IsUseRecord><!--是否启用录像：0-否，1-是--></IsUseRecord>
      <RecordTime><!--延录时长：0-5s，1-10s，2-30s，3-60s，4-120s，5-300s，6-600s-->
    </RecordTime>
      <PreRecordTime>
        <!--预录时长：0-不预录，1-5s，2-10s，3-15s，4-20s，5-25s，6-30s，7-无限制-->
      </PreRecordTime>
      <RecorderDuration>
        <!--录像视频的最大保存时间（只读），该时间的取值范围为1到60天，0表示长期保存-->
      </RecorderDuration>
      <RedundancyRec><!--是否为冗余视频（只读）：0-否，1-是--></RedundancyRec>
      <AudioRec><!--是否录制音频数据（只读）：0-否，1-是--></AudioRec>
      <RECORDPLAY>
        <IsAllDayRecord><!--是否处于全天录像模式：0-否，1-是--></IsAllDayRecord>
        <AllDayRecordType>
          <!--录像触发类型：0-计划录像，1-移动侦测，2-报警，3-移动侦测或报警，4-移动侦测和报警，5-命令，6-手动，7-震动报警，8-场景变更报警，9-VCA报警，10-PIR报警，11-无线报警，12-紧急报警，12-所有报警-->
        </AllDayRecordType>
        <TIME><!--录像计划中的时间段-->
          <RecordType><!--录像触发类型，与AllDayRecordType相同--></RecordType>
          <StartTime><!--开始时间，例如：12:00--></StartTime>
          <StopTime><!--结束时间，例如：12:00--></StopTime>
        </TIME>
        <TIME>
          <RecordType></RecordType>
          <StartTime></StartTime>
          <StopTime></StopTime>
        </TIME>
      </RECORDPLAY>
    </RECORDTIMECFG>
  </ConfigXML>
</Params>

```

输出参数（pOutBuf）

```

<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>

```

```

<Version>4.0</Version>
<Sequence>1</Sequence>
<CommandType>RESPONSE</CommandType>
<WhichCommand>SETDEVICECONFIG</WhichCommand>
<Status>200</Status>
<Description>OK</Description>
</PPVSPMessage>

```

16.2.37 SetRS232Para

设置 RS-232 串口参数。

需调用的接口和命令 (pCmdBuf)

接口：NET_ECMS_XMLConfig

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>SetRS232Para</ConfigCmd>
  <ConfigXML>
    <RS232INFO>
      <BaudRate>
        <!--波特率：0-50, 1-75, 2-110, 3-150, 4-300, 5-600, 6-1200, 7-2400, 8-4800,
        9-9600, 10-19200, 11-38400, 12-57600, 13-76800, 14-115.2K-->
      </BaudRate>
      <DataBit><!--数据位个数：0-5 位, 1-6 位, 2-7 位, 3-8 位--></DataBit>
      <StopBit><!--停止位个数：0-1 位, 1-2 位--></StopBit>
      <Parity><!--校验类型：0-无, 1-奇校验, 2-偶校验--></Parity>
      <FlowControl><!--流控制类型：0-无, 1-软件流控制, 2-硬件流控制--></FlowControl>
      <WorkMode><!--工作模式：0-用于 PPP 拨号, 1-用于参数控制, 2-作为透明通道--></
WorkMode>
    </RS232INFO>
  </ConfigXML>
</Params>

```

输出参数 (pOutBuf)

```

<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>

```

16.2.38 SetRS485Para

设置 RS-485 串口参数。

需调用的接口和命令（pCmdBuf）

接口：NET_ECMS_XMLConfig

命令（pCmdBuf）：SETDEVICECONFIG

输入参数（pInBuf）

```
<Params>
  <ConfigCmd>SetRS485Para</ConfigCmd>
  <ConfigParam1><!--通道号，从1开始--></ConfigParam1>
  <ConfigXML>
    <RS485INFO>
      <BaudRate>
        <!--波特率：0-50，1-75，2-110，3-150，4-300，5-600，6-1200，7-2400，8-4800，
        9-9600，10-19200，11-38400，12-57600，13-76800，14-115.2K-->
      </BaudRate>
      <DataBit><!--数据位个数：0-5位，1-6位，2-7位，3-8位--></DataBit>
      <StopBit><!--停止位个数：0-1位，1-2位--></StopBit>
      <Parity><!--校验类型：0-无，1-奇校验，2-偶校验--></Parity>
      <FlowControl><!--流控制类型：0-无，1-软件流控制，2-硬件流控制--></FlowControl>
      <DecoderType><!--解码器协议类型--></DecoderType>
      <DecoderAddr><!--解码器地址，取值范围为0到255--></DecoderAddr>
    </RS485INFO>
  </ConfigXML>
</Params>
```

输出参数（pOutBuf）

```
<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>
```

16.2.39 SetVILostPara

设置视频丢失侦测参数。

需调用的接口和命令 (pCmdBuf)

接口：[NET_ECMS_XMLConfig](#)

命令 (pCmdBuf)：SETDEVICECONFIG

输入参数 (pInBuf)

```

<Params>
  <ConfigCmd>SetVILostPara</ConfigCmd>
  <!--通道号, 从 1 开始-->
  <ConfigParam1></ConfigParam1>
  <ConfigXML>
    <VILOSTCFG>
      <IsUseVILost><!--是否启用视频丢失侦测: 0-否, 1-是--></IsUseVILost>
    </VILOSTCFG>
  </ConfigXML>
</Params>

```

输出参数 (pOutBuf)

```

<?xml version="1.0" encoding="UTC-8"?>
<PPVSPMessage>
  <Version>4.0</Version>
  <Sequence>1</Sequence>
  <CommandType>RESPONSE</CommandType>
  <WhichCommand>SETDEVICECONFIG</WhichCommand>
  <Status>200</Status>
  <Description>OK</Description>
</PPVSPMessage>

```

16.3 请求 URI

16.3.1 /ISAPI/System/capabilities

获取设备能力。

URI 定义

表 16-1 GET /ISAPI/System/capabilities

方法	GET
描述	获取设备能力。
查询参数	无

请求	无
响应	成功： <u><i>XML_DeviceCap</i></u> 失败： <u><i>XML_ResponseStatus</i></u>

16.3.2 /ISAPI/System/consumptionMode/capabilities?format=json

获取设备功耗模式能力。

请求 URI 定义

表 16-2 GET /ISAPI/System/consumptionMode/capabilities?format=json

方法	GET
描述	获取设备功耗模式能力。
查询参数	format : 决定请求或响应报文的格式。
请求	无
响应	成功： <u><i>JSON_Cap_ConsumptionMode</i></u> 失败： <u><i>JSON_ResponseStatus</i></u>

16.3.3 /ISAPI/System/consumptionMode?format=json

获取或设置设备功耗模式参数。

请求 URI 定义

表 16-3 GET /ISAPI/System/consumptionMode?format=json

方法	GET
描述	获取设备功耗模式参数。
查询参数	format : 决定请求或响应报文的格式。
请求	无
响应	成功： <u><i>JSON_ConsumptionMode</i></u>

	失败： <u><i>JSON_ResponseStatus</i></u>
--	---------------------------------------

表 16-4 PUT /ISAPI/System/consumptionMode?format=json

方法	PUT
描述	设置设备功耗模式参数。
查询参数	format : 决定请求或响应报文的格式。
请求	<u><i>JSON_ConsumptionMode</i></u>
响应	<u><i>JSON_ResponseStatus</i></u>

16.4 XML 或 JSON 报文

16.4.1 JSON_Cap_ConsumptionMode

关于设备功耗模式切换能力的 JSON 报文

```
{
  "ConsumptionMode": {
    "enabled": {
      /*必选项，boolean，是否开启功耗模式*/
      "@opt": [true, false]
    },
    "devWorkMode": {
      /*必选项，string，设备功耗模式：“fullPowerConsumption”-全功耗模式，
      “lowPowerConsumption”-低功耗模式，“sleepOrWakeup”-休眠唤醒模式，该模式下，不支持低电休眠
      参数配置，即 Sleep 和 powerThreshold 无效；若设备支持休眠唤醒模式，默认为休眠唤醒模式，若不支
      持，默认为低电休眠模式*/
      "@opt": ["fullPowerConsumption", "lowPowerConsumption", "sleepOrWakeup"]
    },
    "Sleep": {
      "enabled": {
        /*必选项，boolean，是否开启低电休眠，默认开启；当 enabled 值为 true 时，若设备实际电量低于电量
        阈值，设备进入休眠模式；当 enabled 值为 false 时，就算设备实际电量低于休眠电量阈值，设备也不会进
        入休眠模式*/
        "@opt": [true, false]
      }
      "enabledPrompt": {
        /*可选项，string 数组，设备低电休眠提示语：“prompt1”-低电休眠模式下只支持定时唤醒抓拍，
        “prompt2”-低电休眠模式下只支持 pir 唤醒抓拍*/
        "@opt": ["prompt1", "prompt2"]
      }
    }
  }
}
```

```

    }
  },
  "powerThreshold": {
/*必选项, int, 设备待机休眠电量阈值, 默认值为 30 ;单位 : %*/
    "@min":,
    "@max":,
    "@def":30
  }
  "sleepOrWakeupPrompt": {
/*可选项, string 数组, 设备休眠唤醒模式提示语: “prompt1” -休眠唤醒模式下只支持 pir 唤醒抓拍*/
    "@opt": ["prompt1"]
  }
}
}
}

```

16.4.2 JSON_ConsumptionMode

关于功耗模式参数的 JSON 报文

```

{
  "ConsumptionMode":{
    "enabled":,
/*必选项, boolean, 是否开启功耗模式*/
    "devWorkMode": "",
/*必选项, string, 设备功耗模式:“fullPowerConsumption” -全功耗模式,
“lowPowerConsumption” -低功耗模式, “sleepOrWakeup” -休眠唤醒模式, 该模式下, 不支持低电休
眠, 即 Sleep 和 powerThreshold 无效; 若设备支持休眠唤醒模式, 默认为休眠唤醒模式, 若不支持, 默认
为低电休眠模式*/
    "Sleep":{
      "enabled":
/***必选项, boolean, 是否开启低电休眠, 默认开启; 当 enabled 值为 true 时, 若设备实际电量低于电量
阈值, 设备进入休眠模式; 当 enabled 值为 false 时, 就算设备实际电量低于休眠电量阈值, 设备也不会进
入休眠模式*/
    },
    "powerThreshold":
/*必选项, int, 设备待机休眠电量阈值, 默认值为 30 ;单位 : %*/
  }
}
}

```

16.4.3 JSON_DasInfo

关于设备重定向请求参数的 JSON 报文

```

{
  "Type": "DAS",
  "DasInfo": {
    "Address": "",

```

```

/*必选项, 中心管理服务器 (CMS) 的 IP 地址*/
    "Domain": "",
/*可选项, 域名*/
    "ServerID": "",
/*必选项, 中心管理服务器 (CMS) ID*/
    "Port": ,
/*必选项, 中心管理服务器 (CMS) 的端口号*/
    "UdpPort":
/*必选项, 中心管理服务器 (CMS) 的 UDP 端口号*/
}
}

```

16.4.4 JSON_EventNotificationAlert_BatteryStatus

关于电池状态事件详情的 JSON 报文

```

{
    "ipAddress": "",
/*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 位, 例如, 172.6.64.7*/
    "ipv6Address": "",
/*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 位*/
    "portNo": ,
/*可选项, integer32, 报警设备的端口号*/
    "protocol": "",
/*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
传输;其最大取值长度为 32 位*/
    "macAddress": "",
/*可选项, string, MAC 地址, 其最大取值长度为 32 位, 例如, 01:17:24:45:D9:F4*/
    "channelID": ,
/*可选项, integer32, 报警触发通道号*/
    "dateTime": "",
/*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 位, 例如,
2004-05-03T17:30:08+08:00*/
    "activePostCount": ,
/*必选项, integer32, 单个报警上传次数*/
    "eventType": "",
/*必选项, string, 事件类型, 此处需要设置为“batteryStatus”*/
    "eventState": "",
/*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生 (心跳数据);其最大取值长度
为 32 位*/
    "eventDescription": "",
/*必选项, string, 事件描述, 其最大取值长度为 128 位*/
    "deviceID": "",
/*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
    "batteryPercentage": ,
/*可选项, int, 电池电量剩余百分比,“0-100”*/
    "batteryType":
/*必选项, int, 电池类型: 0-干电池, 1-蓄电池*/

```

```

    "remainingBattery": 10,
    /*可选项, int, 剩余电量, 单位:mA*/
    "totalVoltage": 30,
    /*可选项, int, 总电压, 单位:mV*/
    "cellVoltage": 30,
    /*可选项, int, 单体电压, 单位:mV*/
    "current": 30
    /*可选项, int, 电流, 单位:mA*/
}

```

16.4.5 JSON_EventNotificationAlert_FaceCaptureMsg

关于人脸抓拍事件的 JSON 报文（附带二进制或 URL 格式的事件图片）

图片以 URL 格式上传

```

{
    "ipAddress": "",
    /*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 字符, 例如, 172.6.64.7*/
    "ipv6Address": "",
    /*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 字符*/
    "portNo": ,
    /*可选项, integer32, 报警设备的端口号*/
    "protocol": "",
    /*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
    传输;其最大取值长度为 32 字符*/
    "macAddress": "",
    /*可选项, string, MAC 地址, 其最大取值长度为 32 字符, 例如, 01:17:24:45:D9:F4*/
    "channelID": ,
    /*可选项, integer32, 报警触发通道号*/
    "releatedChannelList": [1,2,3],
    /*可选项, array of integers, 报警关联（同源）通道列表（与 channelID 在同一个相机上的通道),
    用于平台收到报警时展示预览或回放*/
    "dateTime": "",
    /*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 字符, 例如,
    2004-05-03T17:30:08+08:00*/
    "activePostCount": ,
    /*必选项, integer32, 单个报警上传次数*/
    "eventType": "",
    /*必选项, string, 事件类型, 此处需要设置为“faceCapture”*/
    "eventState": "",
    /*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生（心跳数据);其最大取值长度
    为 32 字符*/
    "eventDescription": "",
    /*必选项, 事件描述, 其最大取值长度为 128 字符*/
    "channelName": "",
    /*可选项, 通道名称（监控点名称）*/
    "deviceID": "",

```

```
/*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
    "faceCapture": [{
/*array, 人脸抓拍结果信息, 一次识别可以输出多张抓拍图片的结果, 单张图片可能存在多个人脸*/
        "errorCode": ,
/*可选项, int, 人脸图片比对失败时返回的错误码*/
        "errorMsg": "",
/*可选项, string, 人脸图片比对失败时返回的错误信息*/
        "taskId": "",
/*可选项, string, 任务 ID*/
        "traceUuid": "",
/*可选项, string, 目标 ID, 一个人员对应一个目标 ID, 相同的人员多个报警结果对应相同的目标 ID, 最大长度为 32*/
        "traceIdx": ,
/*可选项, integer32, 匹配图片中相似度最高的图片索引号*/
        "panelChannel":2,
/*可选项, int, 关联面板通道*/
        "ruleName":"123",
/*可选项, string, 规则名称*/
        "alarmType":0,
/*可选项, int, ATM 人脸报警类型:0 (异常人脸)、1 (正常人脸)、2 (多张人脸)、3 (墨镜人脸)、4 (打电话人脸)*/
        "targetAttrs":{
/*人脸属性*/
            "deviceId": "",
/*可选项, string, 设备 ID, 最大长度为 64 字符*/
            "deviceChannel": ,
/*可选项, int32, 设备通道号,*/
            "deviceName": "",
/*可选项, string, 设备名称, 最大长度为 128 字符*/
            "faceTime": "",
/*可选项, string, ISO8601 格式的抓拍时间, 最大长度为 64 字符, 例如, 2016-09-10 00:00:00.000*/
            "bkgUrl": ""
/*可选项, string, 背景大图 URL, 最大长度为 256, 例如, "http://absddsdfsdfsdk.jpg" */
            "contentID":"backgroundImage",
/*可选项, 背景大图*/
            "pId": "F4F665D6A18E41308CE9934DCDDD1111",
/*可选项, string, 抓拍的背景图片 ID (内部使用), 最大长度为 32 字符*/
            "thermalURL":
/*可选项, string, 热成像图 URL, 例如, "http://absddsdfsdfsdk.jpg" */
        },
        "faces": [{
/*人脸图片信息数组, 包含识别结果、人脸属性、匹配的图片列表等*/
            "faceId": ,
/*可选项, int32, 人脸缩略图 ID, 此 ID 在单张背景图片中是唯一的*/
            "faceRect":{
/*可选项, 背景图片中的矩形人脸框*/
                "height": ,
                "width": ,
                "x": ,
                "y":
```

```
    },
    "age": {
        "range": ,
        /*可选项, int32, 年龄误差, 如果存在年龄误差, 则年龄为必填,*/
        "value": ,
        "ageGroup": ""
        /*可选项, string, 年龄范围: "unknown" (未知), "child" (儿童), "young" (青年), "middle"
        (中年), "old" (老年), 最大长度为 32 字符*/
    },
    "gender": {
        /*可选项, 性别*/
        "confidence": ,
        /*可选项, float, 置信度*/
        "value": ""
        /*可选项, string, 性别: male-男, female-女, unknown-未知, 最大长度为 32 字符*/
    },
    "glass": {
        /*可选项, 是否戴眼镜*/
        "confidence": ,
        /*可选项, float, 置信度*/
        "value": ""
        /*可选项, string, 是否戴眼镜: no-不戴眼镜, yes-戴眼镜, sunglasses-戴墨镜, 最大长度为 32 字符*/
    },
    "mask": {
        /*可选项, 是否戴口罩*/
        "confidence": ,
        /*可选项, float, 置信度*/
        "value": ""
        /*可选项, string, 是否戴口罩: no-否, yes-是, 最大长度为 32 字符*/
    },
    "smile": {
        /*可选项, 是否微笑*/
        "confidence": ,
        /*可选项, float, 置信度*/
        "value": ""
        /*可选项, string, 是否微笑: no-否, yes-是, 最大长度为 32 字符*/
    },
    "faceExpression": {
        "value": ""
        /*可选项, string, 人脸表情: "unknown" -未知, "surprised" -惊讶, "panic" -害怕, "disgusted"-
        厌恶, "happy" -高兴, "sad" -难过, "angry" -愤怒, "poker-faced" -无表情*/
    },
    "hat": {
        "value": ""
        /*可选项, string, 是否戴帽子: no-否, yes-是, unknown-未知, 最大长度为 32 字符*/
    }
    "URL": "",
    /*可选项, string, 人脸小图 URL, 最大长度为 256*/
    "stayDuration": ,
    /*integer32, 停留在画面中的时间, 单位:毫秒*/
```



```
    "faceScore": ,
/*必选项, integer32, 人脸评分, 范围从 0 到 100*/
    "captureEndMark": ,
/*必选项, boolean, 抓拍结束标记: true-结束, false-未结束 (该节点结合 traceUuid 使用, 表示该
ID 对应的抓拍结束, 并提取识别图片中清晰度最高的图片) */
    "targetID": ""
/*可选项, string, 人脸与人体关联 ID*/
    "FacePictureRect": {
/*可选项, 人脸矩形框, 该坐标为人脸小图(头肩照)中人脸的坐标, 即 faceImage*/
        "height": ,
/*必选项, float, 高度*/
        "width": ,
/*必选项, float, 宽度*/
        "x": ,
/*必选项, float, 左上角顶点的 X 坐标*/
        "y":
/*必选项, float, 左上角顶点的 Y 坐标*/
    },
    "blockingState": "",
/*可选项, string, 目标人脸遮挡状态: "noBlocking" -无遮挡,
"instantaneousInsignificantBlocking" -瞬时轻度遮挡,
"continuousInsignificantBlocking" -持续轻度遮挡, "significantBlocking" -严重遮挡,
"OSD" -OSD 遮挡, "unknown" -未知*/
    "swingAngle": ,
/*可选项, int, 目标人脸旋转角, 范围从-90 到 90 度*/
    "tiltAngle": ,
/*可选项, int, 目标人脸俯仰角, 范围从-90 到 90 度*/
    "pupilDistance": ,
/*可选项, int, 目标人脸瞳距, 范围: 最小值为 10 像素, 最大值为当前分辨率宽度/1.6*/
    "livenessDetectionStatus": ""
/*可选项, string, 活体检测状态: "liveFace" -真人人脸; "notLiveFace" -非真人人脸;
"detectionFailed" -检测失败; "notEnabled" -未开启活体检测*/
    "enterTime": "",
/*可选项, ISO8601 时间格式, 最佳抓拍的进入时间, 如 "2004-05-03T17:30:08+08:00" */
    "exitTime": ""
/*可选项, ISO8601 时间格式, 最佳抓拍的离开时间, 如 "2004-05-03T17:30:08+08:00" */
    "faceSnapThermometryEnabled": ,
/*可选项, boolean, 人脸抓拍测温使能 (人脸抓拍报警是否支持携带温度信息上报) */
    "currTemperature": "",
/*可选项, float, 人脸温度, 范围: [-20.0,150.0]°C, 精确到小数点后 1 位*/
    "isAbnomalTemperature": ,
/*可选项, boolean, 是否温度异常: true-是, false-否*/
    "thermometryUnit": "",
/*可选项, string, 测温单位: "celsius" -摄氏度, "fahrenheit" -华氏度, "kelvin" -开尔文*/
    "alarmTemperature":
/*可选项, float, 全局预警阈值, 保留后一位小数点*/
    },
    "uid": ""
/*事件上报 ID, 长度为 64 字节, 可由时间 (精确到毫秒) 和随机数组成, 例如,
```

```

“2019052010392202600Kb4s3iQPzLignc1sRUbFT0hR039Y0YzuLt0VLf199yZi” */
  }}
  "URLCertificationType": "digest"
/*可选项，string，图片 URL 认证方式：no-无（针对武汉云存储 URL），digest-摘要认证（针对 DVR 或
NVR 本地存储 URL），最大长度为 32*/
}

```

图片以二进制格式上传

```

Content-Type: multipart/form-data; boundary=MIME_boundary
--MIME_boundary
Content-Type: application/json
Content-Length: 480
{
  "ipAddress": "",
/*必选项，string，报警设备的 IPv4 地址，其最大取值长度为 32 字符，例如，172.6.64.7*/
  "ipv6Address": "",
/*必选项，string，报警设备的 IPv6 地址，其最大取值长度为 128 字符*/
  "portNo": ,
/*可选项，integer32，报警设备的端口号*/
  "protocol": "",
/*可选项，string，协议类型：“HTTP”-用于设备网络 SDK 传输，“HTTPS”，“EHome”-用于 ISUP SDK
传输；其最大取值长度为 32 字符*/
  "macAddress": "",
/*可选项，string，MAC 地址，其最大取值长度为 32 字符，例如，01:17:24:45:D9:F4*/
  "channelID": ,
/*可选项，integer32，报警触发通道号*/
  "releatedChannelList": [1,2,3],
/*可选项，array of integers，报警关联（同源）通道列表（与 channelID 在同一个相机上的通道），
用于平台收到报警时展示预览或回放*/
  "dateTime": "",
/*必选项，string，报警触发时间（ISO 8601 格式），其最大取值长度为 32 字符，例如，
2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
/*必选项，integer32，单个报警上传次数*/
  "eventType": "",
/*必选项，string，事件类型，此处需要设置为“faceCapture”*/
  "eventState": "",
/*必选项，string，事件状态：“active”-已发生，“inactive”-未发生（心跳数据）；其最大取值长度
为 32 字符*/
  "eventDescription": "",
/*必选项，事件描述，其最大取值长度为 128 字符*/
  "channelName": "",
/*可选项，通道名称（监控点名称）*/
  "deviceID": "",
/*可选项，string，设备 ID，在 ISUP SDK 传输过程中，该节点必选*/
  "faceCapture": [{
/*array，人脸抓拍结果信息，一次识别可以输出多张抓拍图片的结果，单张图片可能存在多个人脸*/
  "errorCode": ,

```

```
/*可选项, int, 人脸图片比对失败时返回的错误码*/
    "errorMsg": "",
/*可选项, string, 人脸图片比对失败时返回的错误信息*/
    "traceUuid": "",
/*可选项, string, 目标 ID, 一个人员对应一个目标 ID, 相同的人员多个报警结果对应相同的目标 ID, 最大长度为 32*/
    "traceIdx": ,
/*可选项, integer32, 匹配图片中相似度最高的图片索引号*/
    "panelChannel":2,
/*可选项, int, 关联面板通道*/
    "ruleName":"123",
/*可选项, string, 规则名称*/
    "alarmType":0,
/*可选项, int, ATM 人脸报警类型:0(异常人脸)、1(正常人脸)、2(多张人脸)、3(墨镜人脸)、4(打电话人脸)*/
    "targetAttrs":{
/*人脸属性*/
        "deviceId": "",
/*可选项, string, 设备 ID, 最大长度为 64 字符*/
        "deviceChannel": ,
/*可选项, int32, 设备通道号,*/
        "deviceName": "",
/*可选项, string, 设备名称, 最大长度为 128 字符*/
        "faceTime": "",
/*可选项, string, ISO8601 格式的抓拍时间, 最大长度为 64 字符, 例如, 2016-09-10 00:00:00.000*/
        "contentID": "",
/*可选项, string, 背景图片 ID*/
        "pId": ""
/*可选项, string, 抓拍的背景图片 ID (内部使用), 最大长度为 32 字符*/
    },
    "faces": [{
/*人脸图片信息数组, 包含识别结果、人脸属性、匹配的图片列表等*/
        "faceId": ,
/*可选项, int32, 人脸缩略图 ID, 此 ID 在单张背景图片中是唯一的*/
        "faceRect":{
/*可选项, 背景图片中的矩形人脸框*/
            "height": ,
            "width": ,
            "x": ,
            "y":
        },
        "age":{
            "range": ,
/*可选项, int32, 年龄误差, 如果存在年龄误差, 则年龄为必填,*/
            "value": ,
            "ageGroup": ""
/*可选项, string, 年龄范围: "unknown" (未知), "child" (儿童), "young" (青年), "middle" (中年), "old" (老年), 最大长度为 32 字符*/
        },
        "gender":{
```

```
/*可选项, 性别*/
    "confidence": ,
/*可选项, float, 置信度*/
    "value": ""
/*可选项, string, 性别: male-男, female-女, unknown-未知, 最大长度为 32 字符*/
    },
    "glass": {
/*可选项, 是否戴眼镜*/
    "confidence": ,
/*可选项, float, 置信度*/
    "value": ""
/*可选项, string, 是否戴眼镜: no-不戴眼镜, yes-戴眼镜, sunglasses-戴墨镜, 最大长度为 32 字符*/
    },
    "mask": {
/*可选项, 是否戴口罩*/
    "confidence": ,
/*可选项, float, 置信度*/
    "value": ""
/*可选项, string, 是否戴口罩: no-否, yes-是, 最大长度为 32 字符*/
    },
    "smile": {
/*可选项, 是否微笑*/
    "confidence": ,
/*可选项, float, 置信度*/
    "value": ""
/*可选项, string, 是否微笑: no-否, yes-是, 最大长度为 32 字符*/
    },
    "faceExpression": {
    "value": ""
/*可选项, string, 人脸表情:“unknown” -未知,“surprised” -惊讶,“panic” -害怕,“disgusted”-
厌恶,“happy” -高兴,“sad” -难过,“angry” -愤怒,“poker-faced” -无表情*/
    },
    "hat": {
    "value": ""
/*可选项, string, 是否戴帽子: no-否, yes-是, unknown-未知, 最大长度为 32 字符*/
    }
    "contentID": "",
/*可选项, 人脸小图*/
    "stayDuration": ,
/*integer32, 停留在画面中的时间, 单位:毫秒*/
    "faceScore": ,
/*必选项, integer32, 人脸评分, 范围从 0 到 100*/
    "captureEndMark": ,
/*必选项, boolean, 抓拍结束标记: true-结束, false-未结束 (该节点结合 traceUuid 使用, 表示该
ID 对应的抓拍结束, 并提取识别图片中清晰度最高的图片) */
    "FacePictureRect": {
/*可选项, 人脸矩形框, 该坐标为人脸小图(头肩照)中人脸的坐标, 即 faceImage*/
    "height": ,
/*必选项, float, 高度*/
```

```
        "width": ,
/*必选项, float, 宽度*/
        "x": ,
/*必选项, float, 左上角顶点的 x 坐标*/
        "y":
/*必选项, float, 左上角顶点的 y 坐标*/
    },
    "blockingState": "",
/*可选项, string, 目标人脸遮挡状态:“noBlocking”-无遮挡,
“instantaneousInsignificantBlocking”-瞬时轻度遮挡,
“continuousInsignificantBlocking”-持续轻度遮挡,“significantBlocking”-严重遮挡,
“OSD”-OSD 遮挡,“unknown”-未知*/
    "swingAngle": ,
/*可选项, int, 目标人脸旋转角, 范围从-90 到 90 度*/
    "tiltAngle": ,
/*可选项, int, 目标人脸俯仰角, 范围从-90 到 90 度*/
    "pupilDistance": ,
/*可选项, int, 目标人脸瞳距, 范围:最小值为 10 像素, 最大值为当前分辨率宽度/1.6*/
    "livenessDetectionStatus": ""
/*可选项, string, 活体检测状态:“liveFace”-真人人脸;“notLiveFace”-非真人人脸;
“detectionFailed”-检测失败;“notEnabled”-未开启活体检测*/
    "pId": "",
/*可选项, string, 长度为 32 个字符;人脸子图 ID, 该人脸子图用于显示, 是目标放大后的图片*/
    "enterTime": "",
/*可选项, ISO8601 时间格式, 最佳抓拍的进入时间, 如“2004-05-03T17:30:08+08:00”*/
    "exitTime": ""
/*可选项, ISO8601 时间格式, 最佳抓拍的离开时间, 如“2004-05-03T17:30:08+08:00”*/
    "faceSnapThermometryEnabled": ,
/*可选项, boolean, 人脸抓拍测温使能(人脸抓拍报警是否支持携带温度信息上报)*/
    "currTemperature": "",
/*可选项, float, 人脸温度, 范围: [-20.0,150.0]°C, 精确到小数点后 1 位*/
    "isAbnomalTemperature": ,
/*可选项, boolean, 是否温度异常:true-是, false-否*/
    "thermometryUnit": "",
/*可选项, string, 测温单位:“celsius”-摄氏度,“fahrenheit”-华氏度,“kelvin”-开尔文*/
    "alarmTemperature":
/*可选项, float, 全局预警阈值, 保留后一位小数点*/
    },
    "uid": ""
/*事件上报 ID, 长度为 64 字节, 可由时间(精确到毫秒)和随机数组成, 例如,
“2019052010392202600Kb4s3iQPzLignc1sRUBFT0hR039Y0YzuLt0VLf199yZi”*/
    }
}
--MIME_boundary
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD1111";
filename="faceCapturePicture.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: backgroundImage
```

```
fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD2222";
filename="facePicture.jpg";
Content-Type: image/jpeg
Content-Length: 4385
Content-ID: faceImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD2223";
filename="thermalPicture.jpg";
Content-Type: image/jpeg
Content-Length: 4385
Content-ID: thermalPicture

fefefwageegfqaeg...
--MIME_boundary--
```

16.4.6 JSON_EventNotificationAlert_FaceModelingMsg

关于人脸建模事件的 JSON 报文

人脸建模结果以 URL 形式上传

```
{
  "ipAddress": "",
  /*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 字符, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 字符*/
  "portNo": ,
  /*可选项, integer32, 报警设备的端口号*/
  "protocol": "",
  /*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
传输;其最大取值长度为 32 字符*/
  "macAddress": "",
  /*可选项, string, MAC 地址, 其最大取值长度为 32 字符, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
  /*可选项, integer32, 报警触发通道号*/
  "dateTime": "",
  /*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 字符, 例如,
2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
  /*必选项, integer32, 单个报警上传次数*/
  "eventType": "",
  /*必选项, string, 事件类型, 此处需要设置为“alarmResult”*/
  "eventState": "",
```

```
/*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生(心跳数据);其最大取值长度为32字符*/
  "eventDescription": "",
/*必选项, 事件描述, 其最大取值长度为128字符*/
  "channelName": "",
/*可选项, 通道名称(监控点名称)*/
  "deviceID": "",
/*可选项, string, 设备ID, 在ISUP SDK传输过程中, 该节点必填*/
  "ModelingResult": [{
/*建模结果信息*/
    "targetID": ,
/*必选项, integer类型, 抓拍目标ID*/
    "uid": "",
/*64字符报警上传ID, 由时间(精确到毫秒)和随机数字组成, 例如,
2018070420504791100E4ORyu67pug56nUQ8C6uof48mz97s61mkl3P7C7Hp9u2*/
    "modelingStatus": "",
/*必选项, 建模状态*/
    "Face": {
/*可选项, 人脸属性*/
      "Rect": {
/*可选项, 人脸缩略图的矩形框坐标*/
        "height": ,
/*必选项, float, 人脸框高度*/
        "width": ,
/*必选项, float, 人脸框宽度*/
        "x": ,
/*必选项, float, 人脸框左上角的x坐标*/
        "y":
/*必选项, float, 人脸框左上角的y坐标*/
      },
      "facePicQuality": "",
/*可选项, string, 人脸小图的质量等级:“lower”-低,“medium”-中,“high”-高*/
      "stayDuration": ,
/*可选项, integer32, 画面停留时长, 单字符:毫秒*/
      "faceScore": ,
/*必选项, integer32, 人脸评分, 其取值范围从0到100*/
      "Property": [{
        "description": "age",
        "value":
/*可选项, integer32, 年龄*/
      },
      {
        "description": "ageGroup",
        "value": ""
/*可选项, string, 年龄组:“unknown”-未知,“child”-儿童,“young”-青少年,“middle”-中年,
“old”-老年;最大取值长度为32字符*/
      },
      {
        "description": "gender",
        "value": ""
```

```
/*可选项, string, 性别：“male”-男, “female”-女, “unknown”-未知；最大取值长度为 32 字符*/
},
{
  "description": "glass",
  "value": ""
/*可选项, string, 是否戴眼镜：“no”-否, “yes”-是, “sunglasses”-太阳眼镜；最大取值长度为 32
字符*/
},
{
  "description": "smile",
  "value": ""
/*可选项, string, 是否微笑：“no”-否, “yes”-是；最大取值长度为 32 字符*/
},
{
  "description": "mask",
  "value": ""
/*可选项, string, 是否戴口罩：“no”-否, “yes”-是；最大取值长度为 32 字符*/
},
{
  "description": "faceExpression",
  "value": ""
/*可选项, string, 人脸表情：“unknown”-未知, “poker-faced”-无表情, “happy”-高兴,
“surprised”-惊讶, “panic”-害怕, “sad”-伤心, “angry”-生气, “contemptuous”-轻蔑,
“disgusted”-恶心；最大取值长度为 32 字符*/
},
{
  "description": "hat",
  "value": "unknown"
/*可选项, string, 是否戴帽：“no”-否, “yes”-是, “unknown”-未知；最大取值长度为 32 字符*/
}],
  "modeldata": "",
/*可选项, 由 Base64 编码的建模数据*/
  "faceImageURL": "",
/*人脸小图 URL*/
  "faceBackgroundImageURL": "",
/*背景图片 URL*/
  "FacePictureRect": {
/*可选项, 头肩照中的人脸矩形框坐标*/
    "height": ,
/*必选项, float, 人脸框高度*/
    "width": ,
/*必选项, float, 人脸框宽度*/
    "x": ,
/*必选项, float, 人脸框左上角的 x 坐标*/
    "y":
/*必选项, float, 人脸框左上角的 x 坐标*/
  },
  "livenessDetectionStatus": ""
/*可选项, string, 活体检测状态：“liveFace”-真实人脸, “notLiveFace”-非真实人脸,
“detectionFailed”-检测失败, “notEnabled”-禁用*/
```



```

    }
  ]]
}

```

人脸建模结果以二进制形式上传

```

{
  "ipAddress": "",
  /*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 字符, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 字符*/
  "portNo": ,
  /*可选项, integer32, 报警设备的端口号*/
  "protocol": "",
  /*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
  传输;其最大取值长度为 32 字符*/
  "macAddress": "",
  /*可选项, string, MAC 地址, 其最大取值长度为 32 字符, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
  /*可选项, integer32, 报警触发通道号*/
  "dateTime": "",
  /*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 字符, 例如,
  2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
  /*必选项, integer32, 单个报警上传次数*/
  "eventType": "",
  /*必选项, string, 事件类型, 此处需要设置为“faceSnapModeling”*/
  "eventState": "",
  /*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生 (心跳数据);其最大取值长度
  为 32 字符*/
  "eventDescription": "",
  /*必选项, 事件描述, 其最大取值长度为 128 字符*/
  "channelName": "",
  /*可选项, 通道名称 (监控点名称) */
  "deviceID": "",
  /*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
  "ModelingResult": [{
  /*建模结果信息*/
    "targetID": ,
    /*必选项, integer 类型, 抓拍目标 ID*/
    "uid": "",
    /*64 字符报警上传 ID, 由时间 (精确到毫秒) 和随机数字组成, 例如,
    2018070420504791100E40Ryu67pug56nUQ8C6uof48mz97s61mkl3P7C7Hp9u2*/
    "modelingStatus": "",
    /*必选项, 建模状态*/
    "Face": {
    /*可选项, 人脸属性*/
      "Rect": {
    /*可选项, 人脸缩略图的矩形框坐标*/

```

```
"height": ,
/*必选项, float, 人脸框高度*/
"width": ,
/*必选项, float, 人脸框宽度*/
"x": ,
/*必选项, float, 人脸框左上角的 x 坐标*/
"y":
/*必选项, float, 人脸框左上角的 y 坐标*/
},
"facePicQuality": "",
/*可选项, string, 人脸小图的质量等级:“lower”-低, “medium”-中, “high”-高*/
"stayDuration": ,
/*可选项, integer32, 画面停留时长, 单字符: 毫秒*/
"faceScore": ,
/*必选项, integer32, 人脸评分, 其取值范围从 0 到 100*/
"Property": [{
  "description": "age",
  "value":
/*可选项, integer32, 年龄*/
},
{
  "description": "ageGroup",
  "value": ""
/*可选项, string, 年龄组:“unknown”-未知, “child”-儿童, “young”-青少年, “middle”-中年,
“old”-老年;最大取值长度为 32 字符*/
},
{
  "description": "gender",
  "value": ""
/*可选项, string, 性别:“male”-男, “female”-女, “unknown”-未知;最大取值长度为 32 字符*/
},
{
  "description": "glass",
  "value": ""
/*可选项, string, 是否戴眼镜:“no”-否, “yes”-是, “sunglasses”-太阳眼镜;最大取值长度为 32
字符*/
},
{
  "description": "smile",
  "value": ""
/*可选项, string, 是否微笑:“no”-否, “yes”-是;最大取值长度为 32 字符*/
},
{
  "description": "mask",
  "value": ""
/*可选项, string, 是否戴口罩:“no”-否, “yes”-是;最大取值长度为 32 字符*/
},
{
  "description": "faceExpression",
  "value": ""
```

```

/*可选项, string, 人脸表情:“unknown”-未知,“poker-faced”-无表情,“happy”-高兴,
“surprised”-惊讶,“panic”-害怕,“sad”-伤心,“angry”-生气,“contemptuous”-轻蔑,
“disgusted”-恶心;最大取值长度为 32 字符*/
    },
    {
        "description": "hat",
        "value": "unknown"
    }
/*可选项, string, 是否戴帽:“no”-否,“yes”-是,“unknown”-未知;最大取值长度为 32 字符*/
    }],
    "modeldata": "",
/*可选项, 由 Base64 编码的建模数据*/
    "contentID1": "",
    "contentID2": "",
    "FacePictureRect": {
/*可选项, 头肩照中的人脸矩形框坐标*/
        "height": ,
/*必选项, float, 人脸框高度*/
        "width": ,
/*必选项, float, 人脸框宽度*/
        "x": ,
/*必选项, float, 人脸框左上角的 x 坐标*/
        "y":
/*必选项, float, 人脸框左上角的 x 坐标*/
    },
    "livenessDetectionStatus": ""
/*可选项, string, 活体检测状态:“liveFace”-真实人脸,“notLiveFace”-非真实人脸,
“detectionFailed”-检测失败,“notEnabled”-禁用*/
    },
    "pId": ""
/*string, 人脸小图 ID, 其最大取值长度为 32 字符, 例如, F4F665D6A18E41308CE9934DCDDD2222*/
    }],
    "pId": ""
/*string, 背景图片 ID (内部使用), 其最大取值长度为 32 字符, 例如,
F4F665D6A18E41308CE9934DCDDD1111*/
}
--boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename="faceCapturePicture.jpg"
Content-Type:image/jpeg
Content-Length: 118579
Content-ID: faceImage

<图片数据>
--boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD2222";
filename="facePicture.jpg"
Content-Type: image/jpeg
Content-Length: 124209

```

```
Content-ID: faceBackgroundImage
```

```
<图片数据>
--boundary--
```

16.4.7 JSON_EventNotificationAlert_FacePicComparisonMsg

关于人脸图片比对报警详情的 JSON 报文

人脸比对（图片二进制数据）

```
{
  "ipAddress": "172.6.64.7",
  /*必选项, string, 报警设备的 IPv4 地址, 最大长度为 32 字节*/
  "ipv6Address": "",
  /*可选项, string, 报警设备的 IPv6 地址, 最大长度为 128 字节*/
  "portNo": 80,
  /*可选项, integer32, 报警设备的端口号*/
  "protocolType": "HTTP",
  /*可选项, string, 协议类型: "HTTP"、"HTTPS"。最大长度为 32 字节*/
  "macAddress": "01:17:24:45:D9:F4",
  /*可选项, string, MAC 地址, 最大长度为 32 字节*/
  "channelID": "1",
  /*可选项, integer32, 触发报警的设备通道号*/
  "releatedChannelList": [1,2,3],
  /*可选项, array of integers, 报警关联（同源）通道列表（与 channelID 在同一个相机上的通道），
  用于平台收到报警时展示预览或回放*/
  "dateTime": "2004-05-03T17:30:08+08:00",
  /*必选项, string, 报警触发时间, ISO 8601 时间格式, 最大长度为 32 字节*/
  "activePostCount": 1,
  /*必选项, integer32, 同一个报警已经上传的次数*/
  "eventType": "alarmResult",
  /*必选项, string, 触发的事件类型: "alarmResult" -人脸比对。最大长度为 128 字节*/
  "eventState": "active",
  /*必选项, string, 事件触发状态: "active" -触发、"inactive" -未触发（表示心跳数据）。最大长度
  为 32 字节*/
  "eventDescription": "alarmResult",
  /*必选项, string, 事件描述: "alarmResult" -人脸比对。最大长度为 128 字节*/
  "channelName": "",
  /*可选项, string, 通道名称（监控点名称）*/
  "deviceID": "test0123",
  /*可选项, 设备 ID, 即 PUID, 在 ISUP 协议接入透传 ISAPI 事件信息中必须返回*/
  "alarmResult": [{
  /*array, 比对报警结果, 一次识别可以输出多个人员的比对结果, 单个人员报警识别结果中取相似度最高的
  抓拍图片*/
    "errorCode": 1,
  /*可选项, int, 当人脸图片分析失败时, 会返回对应的错误码*/
    "errorMsg": "ok",
```

```
/*可选项, string, 当人脸图片分析失败时, 会返回对应的错误信息*/
    "modelData":"","
/*可选项, string, 比对任务中的目标模型比对数据, 最大长度为 2048 字节*/
    "traceUuid":"sdddddddddff",
/*可选项, string, 目标 ID, 一个人员对应一个目标 ID, 相同的人员多个报警结果对应相同的目标 ID (对
应/ISAPI/SDT/Management/Task/facePicture 中的 traceUuid 字段)。最大长度为 64 字节*/
    "traceIdx":1,
/*可选项, integer32, 图片索引, 同一个人匹配的多张图片中的索引 (对应/ISAPI/SDT/Management/
Task/facePicture 中的 traceIdx 字段) */
    "targetAttrs":{
/*目标属性*/
        "deviceId":"ee3c4000-a22d-11b4-8334-5803fb77894c",
/*可选项, string, 设备 UUID, 最大长度为 64*/
        "deviceChannel":1,
/*可选项, integer32, 设备通道号*/
        "deviceName":"shebei",
/*可选项, string, 设备名称, 最大长度为 128 字节*/
        "faceTime":"2016-09-10 00:00:00.000",
/*可选项, string, 抓拍时间, 格式例如: 2016-09-10 00:00:00.000, 最大长度为 64 字节*/
        "rect":{
/*可选项, 人脸大图矩形框*/
            "height":1.0,
/*可选项, float, 高度*/
            "width":1.0,
/*可选项, float, 宽度*/
            "x":0.0,
/*可选项, float, X 坐标*/
            "y":0.0
/*可选项, float, Y 坐标*/
        },
        "contentID":"backgroundImage",
        "pId":"F4F665D6A18E41308CE9934DCDDD1111"
/*string, 视频采集抓拍的背景图, pId 表示设备内部抓图的唯一标识, 长度是 32 个字符*/
    },
    "faces":[{
/*array, 人脸信息包括检测结果、属性、建模和匹配名单, 一张图片中的可能多个人脸结果*/
        "faceId":1,
/*可选项, integer32, 人脸 ID, 一张图片中的人脸 ID 不能重复*/
        "faceRect":{
/*可选项, 人脸小图矩形框, 用于参数建模*/
            "height":1.0,
/*可选项, float, 高度*/
            "width":1.0,
/*可选项, float, 宽度*/
            "x":0.0,
/*可选项, float, X 坐标*/
            "y":0.0
/*可选项, float, Y 坐标*/
        },

```

```
    "faceMark":{
      "leftEye":{
/*可选项，左眼位置*/
        "x":0.13,
        "y":0.87
      },
      "rightEye":{
/*可选项，右眼位置*/
        "x":0.78,
        "y":0.50
      },
      "noseTip":{
/*可选项，鼻尖位置*/
        "x":0.78,
        "y":0.50
      },
      "leftMouth":{
/*可选项，左嘴角位置*/
        "x":0.78,
        "y":0.50
      },
      "rightMouth":{
/*可选项，右嘴角位置*/
        "x":0.78,
        "y":0.50
      }
    },
    "recommendFaceRect":{
/*可选项，人脸区域推荐位置，用于抠图用，在人脸小图矩形框基础上向外放大。人脸小图矩形框只能刚刚好
标定出人脸，不适合作为抠图展示*/
      "height":1.0,
      "width":1.0,
      "x":0.0,
      "y":0.0
    },
    "facePose":{
/*可选项，人脸角度*/
      "pitch":0.998,
/*可选项，float，平面外上下俯仰角，人脸朝上为正，取值范围：[-90, 90]，若无效时此项不存在，保
留3位有效数字*/
      "yaw":0.876,
/*可选项，float，平面外左右偏转角，人脸朝左为正，取值范围：[-90, 90]，若无效时此项不存在，保
留3位有效数字*/
      "roll":0.623
/*可选项，float，平面内旋转角，人脸顺时针旋转为正，取值范围：[-90, 90]，若无效时此项不存在，
保留3位有效数字*/
    },
    "age":{
/*可选项，年龄*/
      "range":5,

```

```
/*可选项, integer32, 年龄偏差, 年龄偏差值存在时年龄值必选项*/
    "value":23,
/*可选项, integer32, 年龄*/
    "ageGroup":"old"
/*可选项, string, 年龄段, 最大长度为 32 字节*/
    },
    "gender":{
/*可选项, 性别*/
    "confidence":99.99990,
/*可选项, float, 置信度*/
    "value":"male "
/*可选项, string, 性别:“male”-男、“female”-女、“unknown”-未知。最大长度为 32 字节*/
    },
    "glass":{
/*可选项, 是否戴眼镜*/
    "confidence":99.94499999999999,
/*可选项, float, 置信度*/
    "value":"yes"
/*可选项, string, 是否戴眼镜: “no”-不戴眼镜、“yes”-戴眼镜、“sunglasses”-戴墨镜。最大长度为 32 字节*/
    },
    "smile":{
/*可选项, 是否微笑*/
    "confidence":99.99990,
/*可选项, float, 置信度*/
    "value":"yes"
/*可选项, string, 是否微笑: “no”-不微笑、“yes”-微笑。最大长度为 32 字节*/
    },
    "mask":{
/*可选项, 是否戴口罩*/
    "confidence":99.94499999999999,
/*可选项, float, 置信度*/
    "value":""
    },
    "faceExpression":{
/*可选项, 表情*/
    "confidence":99.99990,
/*可选项, float, 置信度*/
    "value":"unknown"
    },
    "hat":{
/*可选项, 帽子*/
    "confidence":99.99990,
/*可选项, float, 置信度*/
    "value":"unknown"
    },
    "score":{
/*可选项, 人脸评分*/
    "confidence":99.99990,
/*可选项, float, 置信度*/
```

```
        "value":65
    },
    "modeldata":"","
/*可选项, string, 模型 Base64 编码数据*/
    "AbsoluteHigh":{
/*可选项, PTZ 位置*/
    "elevation":11,
/*可选项, int, 仰角, 取值范围: [-900,2700]*/
    "azimuth":22,
/*可选项, int, 方位角, 取值范围: [0,3600]*/
    "absoluteZoom":33
/*可选项, int, 变倍倍率, 取值范围: [1,1000]*/
    },
    "identify":[{
        "relationId":"9887",
/*可选项, string, 布防的关联关系 ID (对应/ISAPI/SDT/Management/Task/facePicture 中的
relateGroup 数组中的 id 节点), 来源于用户提交任务是指定的布防比对策略的 ID, 最大长度为 64 字节*/
        "maxsimilarity":0.78,
/*可选项, float, 所有匹配人员比对结果中的最大相似度, 取值范围: [0,1], 精确到小数点后两位*/
        "similarityRange":1,
/*可选项, int, 相似度范围: 0-低于最小值、1-介于最小值和最大值之间、2-大于最大值。对于小于最小值
的人员, 上层会进行打标签处理*/
        "candidate":[{
/*array, 匹配的人员名单信息列表, 同一个布防关联对应多个匹配人员*/
            "alarmId":1,
/*可选项, integer32, 报警 ID, 一个布控关系内不能重复*/
            "blacklist_id":"2323",
/*可选项, string, 报警名单 ID, 对应人脸库中的 FDID, 最大长度为 64; 前端只返回设备默认的 FDID,
自定义人脸库 ID 在 customFaceLibID 中返回*/
            "human_data":[{
/*array, 匹配的人脸信息列表, 同一个人的多张人脸*/
                "face_id":"123445",
/*可选项, string, 人脸 ID, 最大长度为 32 字节*/
                "face_rect":{
                    "height":1.0,
                    "width":1.0,
                    "x":0.0,
                    "y":0.0
                },
                "similarity":0.997000,
/*比对相似度 (一个人员可能会对应多张人脸图片) */
                "isNoSaveFDPicture":true,
/*不保存人脸库图片, 若开启了导入图片或者建模时不保存原图功能, 则需返回该节点为 true, 未开启则无
需返回该节点*/
                "contentID":"faceLibImage",
                "pId":"F4F665D6A18E41308CE9934DCDDD766F"
/*string, 人脸库中的图片 ID, 长度是 32 个字符, 该字段匹配 HTTP 表单格式中文件名称的定义*/
            }],
            "human_id":"123",
/*可选项, string, 名单人员 ID, 对应人脸库中 FPID, 最大长度为 64 字节*/
```



```
"reserve_field":{
/*可选项, 保留字节 (透传数据), 最大不超过 256 字节, 若下发的数据超过此会产生截断*/
    "name":"张三",
/*必选项, string, 姓名, 最大长度为 96 字节*/
    "gender":"male",
/*可选项, string, 性别:“male”-男、“female”-女、“unknown”-未知。最大长度为 10 字节*/
    "bornTime":"2004-05-03",
/*必选项, string, 出生日期, ISO 8601 时间格式, 最大长度为 32 字节*/
    "city":"130100",
/*可选项, string, 人员出生城市编号, 最大长度为 32 字节*/
    "certificateType ":"officerID",
/*可选项, string, 证件类型:“officerID”-军官证、“ID”-身份证、“passport”-护照、“other”-其他。最大长度为 10 字节*/
    "certificateNumber":"",
/*可选项, string, 证件号, 最大长度为 32 字节*/
    "ageGroup":"old",
/*可选项, string, 年龄段, 最大长度为 32 字节*/
    "phoneNumber":""
/*可选项, string, 电话号码, 最大长度为 64 字节*/
    "cityName": "石家庄市",
/*可选项, string, 城市名称 (与 city 字段对应), range:[0,32], desc:使用场景:客户端等计算能力较弱,不能实时根据 city 的编号查询出对应的城市名称*/
    "provinceName": "河北省"
/*可选项, string, 省份名称, range:[0,32], desc:使用场景:客户端等计算能力较弱,不能实时根据 city 的编号查询出对应的省份名称*/
},
"similarity":0.780,
/*必选项, float, 该人员所有匹配人脸图片结果中的最大相似度, 取值范围:[0,1], 精确到小数点后两位*/
"listType":"blackList",
/*可选项, string, 名单类型:“blackList”-黑名单、“whiteList”-白名单。最大长度为 20 字节*/
"extendData":[{
/*可选项, 人员扩展信息*/
    "extendID":1,
/*可选项, int, 人员扩展信息 ID, 从 1 开始*/
    "enable":true,
/*可选项, boolean, 是否启用人员扩展信息功能*/
    "name":"",
/*可选项, string, 人员标签信息扩展名称, 最大长度为 96 字节*/
    "value":""
/*可选项, string, 人员标签信息扩展内容, 最大长度为 64 字节*/
}],
"FDLibName":"faceLib1",
/*可选项, 人脸库名称*/
"FDLibThreshold":50,
/*可选项, 人脸库检测阈值, 取值范围:[0,100]*/
"customFaceLibID":"",
/*可选项, string, 自定义人脸库 ID*/
"customHumanID":""
/*可选项, string, 自定义人员 ID*/
```

```

    }],
    "alarmRecordID": "",
/*可选项, string, 报警记录 ID*/
    "libAttribute": "general"
/*可选项, 库属性类型: "general" -普通库、"blackList" -黑名单库、"VIP" -VIP 库、"passerby"-
路人库*/
    }],
    "contentID": "faceImage",
    "pId": "F4F665D6A18E41308CE9934DCDDD2222",
/*string, 人脸子图图片 ID, 长度是 64 个字符, 该人脸子图用于显示, 是目标识别图扩大后的图片*/
    "stayDuration": 20,
/*可选项, integer32, 停留画面中时间, 单位: 毫秒*/
    "prompt": "",
/*可选项, string, 人脸库比对提示语, 最大长度为 16 个字节*/
    "contrastStatus": true,
/*可选项, boolean, 比对结果*/
    "realTimeContrast": true,
/*可选项, boolean, 是否实时比对*/
    "UUPID": "",
/*可选项, string, 人脸库底库图片唯一标识 ID, 长度最大为 64 位字符串 (序列号+入库时间+随机数) */
    "frontMatchType": 1,
/*可选项, int, 比对类型: 1-聚类库、0-普通库。用于判断比对源是否是聚类库*/
    "isImportFD": true
/*可选项, boolean, 抓拍人脸数据是否入库: true-入库、false-不入库) */
    }],
    "alarmRecordID": "",
/*可选项, string, 报警记录 ID*/
    "uid": "",
/*事件上报的唯一标识, 64 字节的长度, 可以使用时间 (精确到毫秒) 加上随即数的方式组成*/
    "remainReportNum": 0
/*可选项, int, 当前目标比对报警的剩余上报次数*/
    }]
}
--MIME_boundary
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD1111";
filename="faceCapturePicture.jpg";
Content-Type: image/jpeg
Content-Length: 124209
Content-ID: backgroundImage

<背景图片数据>
--MIME_boundary
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD766F";
filename="faceMatchPicture.jpg";
Content-Type: image/jpeg
Content-Length: 51051
Content-ID: faceLibImage

<人脸库图片数据>
--MIME_boundary

```

```
Content-Disposition: form-data; name="F4F665D6A18E41308CE9934DCDDD2222";
filename="facePicture.jpg";
Content-Type: image/jpeg
Content-Length: 124209
Content-ID: faceImage
```

<人脸子图图片数据>
--MIME_boundary--

人脸比对（URL）

```
{
  "ipAddress": "",
  /*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 字符, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 字符*/
  "portNo": ,
  /*可选项, integer32, 报警设备的端口号*/
  "protocol": "",
  /*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
  传输;其最大取值长度为 32 字符*/
  "macAddress": "",
  /*可选项, string, MAC 地址, 其最大取值长度为 32 字符, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
  /*可选项, integer32, 报警触发通道号*/
  "releatedChannelList": [1,2,3],
  /*可选项, array of integers, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道),
  用于平台收到报警时展示预览或回放*/
  "dateTime": "",
  /*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 字符, 例如,
  2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
  /*必选项, integer32, 单个报警上传次数*/
  "eventType": "",
  /*必选项, string, 事件类型, 此处需要设置为“alarmResult”*/
  "eventState": "",
  /*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生 (心跳数据);其最大取值长度
  为 32 字符*/
  "eventDescription": "",
  /*必选项, 事件描述, 其最大取值长度为 128 字符*/
  "alarmResult": [{
  /*人脸图片比对报警信息数组, 若包含多张人脸图片的比对结果, 则选取相似度最高的图片*/
    "errorCode": ,
    /*可选项, 人脸图片比对失败时返回的错误码*/
    "errorMsg": "",
    /*可选项, 人脸图片比对失败时返回的错误信息*/
    "image": "",
    /*可选项, string, 上传的用于比对的人脸图片 URL, 最大长度为 256 字符, 只有当 errorCode=="1"且
    errorMsg=="ok"时有效, 例如, http://10.7.52.174:8080/kms/services/rest/
```

```
dataInfoService/downloadFile?id=00000001/
temp001/011_163957_23006&token=7a57a5a7fffffffffc1a0316369671314*/
  "modelData": "",
/*可选项, string, 目标建模数据, 最大长度为 2048 字符*/
  "traceUuid": "",
/*可选项, string, 目标 ID, 多个报警中, 同一张人脸对应一个 ID, 最大长度为 64 字符*/
  "traceIdx": ,
/*可选项, int32, 匹配的人脸图片索引号*/
  "targetAttrs":{
/*人脸属性*/
    "linkFaceBodyId": "",
/*可选项, 人脸人体关联 ID (HMS、HMSZA) */
    "linkFaceVehicleId": "",
/*可选项, 人脸车辆关联 ID (HMS、卡口) */
    "deviceId": "",
/*可选项, string, 设备 ID, 最大长度为 64 字符*/
    "deviceChannel": ,
/*可选项, int32, 设备通道号,*/
    "deviceName": "",
/*可选项, string, 设备名称, 最大长度为 128 字符*/
    "faceTime": "",
/*可选项, string, ISO8601 格式的抓拍时间, 最大长度为 64 字符, 例如, 2016-09-10 00:00:00.000*/
    "rect":{
/*可选项, 人脸图片的矩形框*/
      "height": ,
/*可选项, float, 矩形框的高度*/
      "width": ,
/*可选项, float, 矩形框的宽度*/
      "x": ,
/*可选项, float, 矩形框的 X 坐标*/
      "y":
/*可选项, float, 矩形框的 Y 坐标*/
    },
    "bkgUrl": "",
/*可选项, string, 背景图片 URL, 最大长度为 256 字符, 同一张背景图片中可能存在多张人脸缩略图, 例如, “http://absddsdfdsfsdk.jpg” */
  },
  "faces": [{
/*人脸图片信息数组, 包含识别结果、人脸属性、匹配的图片列表等*/
    "faceId": ,
/*可选项, int32, 人脸缩略图 ID, 此 ID 在单张背景图片中是唯一的*/
    "faceRect":{
/*可选项, 背景图片中的矩形人脸框*/
      "height": ,
      "width": ,
      "x": ,
      "y":
    },
    "faceMark":{
      "leftEye":{
```

```
/*可选项, 人脸图片中左眼字符置坐标*/
    "x": ,
    "y":
},
"rightEye":{
/*可选项, 人脸图片中右眼字符置坐标*/
    "x": ,
    "y":
},
"noseTip":{
/*可选项, 人脸图片中鼻尖字符置坐标*/
    "x": ,
    "y":
},
"leftMouth":{
/*可选项, 人脸图片中左嘴角字符置坐标*/
    "x": ,
    "y":
},
"rightMouth":{
/*可选项, 人脸图片中右嘴角字符置坐标*/
    "x": ,
    "y":
}
},
"recommendFaceRect":{
/*可选项, 人脸框建议高宽和坐标信息*/
    "height": ,
    "width": ,
    "x": ,
    "y":
},
"facePose":{
/*可选项, 人脸姿态*/
    "pitch": ,
/*可选项, float, 人脸抬头或低头的角度, 范围: [-90,90], 人脸抬头角度为正值, 精确到小数点后三字符。如果无效则该节点不存在*/
    "yaw": ,
/*可选项, float, 人脸左右倾斜的角度, 范围: [-90,90], 人脸向左倾斜角度为正值, 精确到小数点后三字符。如果无效则该节点不存在*/
    "roll": ,
/*可选项, float, 人脸顺时针倾斜或逆时针偏转角度, 范围: [-90,90], 人脸顺时针倾斜角度为正值, 精确到小数点后三字符。如果无效则该节点不存在*/
    },
    "age":{
        "range": ,
/*可选项, int32, 年龄误差, 如果存在年龄误差, 则年龄为必填,*/
        "value": ,
        "ageGroup": ""
/*可选项, string, 年龄范围: "unknown" (未知), "child" (儿童), "young" (青年), "middle"
```

```
(中年) , "old" (老年), 最大长度为 32 字符*/
},
"gender":{
  "confidence": ,
  "value": ""
},
"glass":{
/*可选项, 是否戴眼镜*/
  "confidence": ,
  "value": ""
},
"smile":{
/*可选项, 是否微笑*/
  "confidence": ,
  "value": ""
},
"identify": [{
  "relationId": "",
/*可选项, string, 比对策略 ID, 最大长度为 64 字符*/
  "maxsimilarity": ,
/*可选项, float, 匹配结果中的最大相似度, 相似度取值范围为 0 到 1, 精确到小数点后两字符*/
  "candidate": [{
/*匹配的人脸图片信息数组*/
  "alarmId": ,
/*可选项, int32, 报警 ID, 该 ID 在单次连接中不可重复*/
  "blacklist_id": "",
/*可选项, string, 人脸图片列表 ID, 与人脸库中的 FDID 对应, 最大长度为 64 字符*/
  "human_data": [{
/*匹配人脸图片关联的人员信息数组*/
  "face_id": "",
/*必选项, string, 人脸图片 ID*/
  "face_picurl": "",
/*可选项, string, 人脸缩略图 URL, 例如, http://127.0.0.1:6120/face_picurl*/
  "face_rect":{
    "height": ,
    "width": ,
    "x": ,
    "y":
  },
  "similarity":
/*可选项, float, 人脸图片相似度, 取值范围为 0 到 1*/
  }],
  "human_id": "",
/*可选项, string, 列表中的人员 ID, 与人脸库中的 FPID 对应, 最大长度为 64 字符*/
  "reserve_field":{
/*可选项, 匹配的人员信息, 最大长度为 256 字符*/
    "name": "",
    "gender": "",
    "bornTime": "",
    "city": "",
```

```
        "certificateType": "",
/*可选项, string, 身份凭证类型: "officerID" (警官证), "ID" (身份证), "passport" (护照),
"other" (其他), 最大长度为10字符*/
        "certificateNumber": "",
        "ageGroup": "",
        "phoneNumber": "",
        "cityName": "石家庄市",
/*可选项, string, 城市名称 (与 city 字段对应), range:[0,32], desc:使用场景:客户端等计算能力较弱,不能实时根据 city 的编号查询出对应的城市名称*/
        "provinceName": "河北省"
/*可选项, string, 省份名称, range:[0,32], desc:使用场景:客户端等计算能力较弱,不能实时根据 city 的编号查询出对应的省份名称*/
    },
    "similarity": 0.780,
    "listType": "blackList",
/*可选项, string, 列表类型: "blacklist" (黑名单), "whitelist" (白名单), 最大长度为20字符*/
    "extendData": [{
/*可选项, 关联人员的扩展信息*/
        "extendID": "",
        "enable": "",
        "name": "",
        "value": "",
    }],
    }],
    "URL": "",
/*可选项, string, 人脸图片 URL, 最大长度为256字符*/
    "stayDuration": ,
/*可选项, int32, 人脸在画面中的停留时间, 单字符: 毫秒*/
    "prompt": "",
/*可选项, string, 人脸库比对提示语, 最大长度为16字节*/
    "contrastStatus": true,
/*可选项, boolean, 比对结果*/
    "realTimeContrast": true,
/*可选项, boolean, 是否实时比对*/
    "UUPID": "",
/*可选项, string, 人脸库底库图片唯一标识 ID, 长度最大为64位字符串 (序列号+入库时间+随机数) */
    "frontMatchType": 1,
/*可选项, int, 比对类型: 1-聚类库、0-普通库。用于判断比对源是否是聚类库*/
    "isImportFD": true
/*可选项, boolean, 抓拍人脸数据是否入库: true-入库、false-不入库*/
    }],
    "URLCertificationType": ""
/*可选项, string, 图片 URL 认证模式: "no"-未认证, "digest"-摘要认证, 最大长度为32字符*/
}
```

16.4.8 JSON_EventNotificationAlert_HeatMapMsg

关于热度图上传报告的 JSON 报文

```
{
  "ipAddress": "",
  /*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 位, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 位*/
  "portNo": ,
  /*可选项, integer32, 报警设备的端口号*/
  "protocol": "",
  /*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
  传输;其最大取值长度为 32 位*/
  "macAddress": "",
  /*可选项, string, MAC 地址, 其最大取值长度为 32 位, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
  /*可选项, integer32, 报警触发通道号*/
  "releatedChannelList": [1,2,3],
  /*可选项, array of integers, 报警关联(同源)通道列表(与 channelID 在同一个相机上的通道),
  用于平台收到报警时展示预览或回放*/
  "dateTime": "",
  /*必选项, string, 报警触发时间(ISO 8601 格式), 其最大取值长度为 32 位, 例如,
  2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
  /*必选项, integer32, 单个报警上传次数*/
  "eventType": "",
  /*必选项, string, 事件类型, 此处需要设置为“heatmap”*/
  "eventState": "",
  /*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生(心跳数据);其最大取值长度
  为 32 位*/
  "eventDescription": "",
  /*必选项, 事件描述, 其最大取值长度为 128 位*/
  "channelName": "",
  /*可选项, 通道名称(监控点名称)*/
  "deviceID": "",
  /*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
  "HeatMapValueStatistics":{
  /*必选项, 热度图统计*/
    "startTime": "",
    /*必选项, string, 统计开始时间(ISO 8601 格式), 例如, 2018-04-24T16:56:43.000+08:00*/
    "endTime": "",
    /*必选项, string, 统计结束时间(ISO 8601 格式), 例如, 2018-04-24T16:56:43.000+08:00*/
    "statisticsType": "",
    /*必选项, string, 统计类型:“PDC”-按人员数量统计,“DURATION”-按停留时长统计,
    “INTERSECTION”-按路口人员数量统计*/
    "HeatMapValue": [{
  /*必选项, 结构化热度图信息列表*/
```



```

    "heatmapDataType": "",
    /*可选项, string, 热度图类型: "PDC_stayNum" - 人员停留时长热度图, "PDC_leaveNum" - 离开人员
    热度图, 该节点只有当统计类型 (statisticsType) 为 "PDC" 时有效*/
    "maxHeatMapValue": ,
    /*必选项, integer32, 最大热度值*/
    "minHeatMapValue": ,
    /*必选项, integer32, 最小热度值*/
    "timeHeatMapValue": ,
    /*必选项, integer32, 平均热度值*/
    "lineValue": ,
    /*必选项, integer32, 热度图中每行平均像素值*/
    "columnValue": ,
    /*必选项, integer32, 热度图中每列平均像素值*/
    "curNumber": ,
    /*可选项, int, 当前人员数量, 该节点只有当统计类型 (statisticsType) 为 "PDC" 时有效*/
    "leaveNumber": ,
    /*可选项, int, 离开人员数量, 该节点只有当统计类型 (statisticsType) 为 "PDC" 时有效*/
    "totalTime":
    /*可选项, int, 总停留时长, 单位: 秒, 该节点只有当统计类型 (statisticsType) 为 "DURATION" 时有效*/
    }],
    "heatmapURL": "",
    /*热度图像素点数据 URL, 该节点在统计类型 (statisticsType) 为 "DURATION" 和 "INTERSECTION"
    时有效*/
    "PDC_stayURL": "",
    /*停留人数热度图像素点数据 URL, 该节点只有在统计类型 (statisticsType) 为 "PDC" 时有效*/
    "PDC_leaveURL": ""
    /*离开人数热度图像素点数据 URL, 该节点只有在统计类型 (statisticsType) 为 "PDC" 时有效*/
    }
}

```

16.4.9 JSON_EventNotificationAlert_mixedTargetDetection

关于混合目标检测报警详情的 JSON 报文

二进制格式图片上传

```

Content-Type: multipart/form-data; boundary=MIME_boundary
--MIME_boundary
Content-Type: application/json
Content-Length: 480

{
  "ipAddress": "",
  /*必选项, string, 报警设备 IPv4 地址, 最大长度为 32 字节, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备 IPv6 地址, 最大长度为 128 字节*/
  "portNo": ,
  /*可选项, integer32, 报警设备端口编号*/

```

```
"protocol": "",
/*可选项, string, 协议类型:“HTTP”-设备网络 SDK,“HTTPS”,“EHome”-ISUP SDK;最大长度为 32
字节*/
"macAddress": "",
/*可选项, string, 物理地址, 最大长度为 32 字节, 例如, 01:17:24:45:D9:F4*/
"channelID": ,
/*可选项, integer32, 报警触发通道编号*/
"relatedChannelList": [1,2,3],
/*可选项, array of integers, 报警关联(同源)通道列表(与 channelID 在同一个相机上的通道),
用于平台收到报警时展示预览或回放*/
"dateTime": "",
/*必选项, string, 报警触发时间(ISO 8601 格式), 最大长度为 32 字节, 例如,
2004-05-03T17:30:08+08:00*/
"activePostCount": ,
/*必选项, integer32, 一次报警上传时间*/
"eventType": "",
/*可选项, string, 出现事件类型, 此处应设为“mixedTargetDetection”;最大长度为 128 字节*/
"eventState": "",
/*必选项, string, 事件状态:“active”-已出现,“inactive”-未出现(心跳数据);最大长度为 32 字节*/
"eventDescription": "",
/*必选项, 事件描述, 最大长度为 128 字节*/
"channelName": "",
/*必选项, string, 通道或相机名称*/
"deviceID": "",
/*可选项, string, ISUP SDK 集成期间设备 ID 须返回*/
"isDataRetransmission":,
/*可选项, boolean, 数据重传标记*/
"CaptureResult": [{
/*抓拍和检测结果*/
"targetID": ,
/*必选项, int, 抓拍目标 ID*/
"Face": {
/*可选项, 人脸属性*/
"Rect": {
/*可选项, 人脸缩略图帧信息*/
"height": ,
/*必选项, float, 帧高度*/
"width": ,
/*必选项, float, 帧宽度*/
"x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
"y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
},
"Property": [{
/*人脸属性详情*/
"description": "",
"value": ""
},
},
```

```
{
    "description": "ageGroup",
    "value": ""
/*可选项, string, 年龄段:“unknown (未知)”, “child (少年)”, “young (青年)”, “middle (中年)”,
“old (老年)” ;最大长度为 32 字节*/
    },
    {
        "description": "gender",
        "value": ""
/*可选项, string, 性别:“male(男性)”, “female (女性)”, “unknown (未知)” ;最大长度为 32 字节*/
    },
    {
        "description": "glass",
        "value": ""
/*可选项, string, 是否戴眼镜:“no (否)”, “yes (是)”, “sunglasses (太阳镜)” ;最大长度为 32 字节*/
    },
    {
        "description": "smile",
        "value": ""
/*可选项, string, 是否微笑:“no (否)”, “yes (是)” ;最大长度为 32 字节*/
    },
    {
        "description": "mask",
        "value": ""
/*可选项, string, 是否戴口罩:“no (否)”, “yes (是)” ;最大长度为 32 字节*/
    },
    {
        "description": "faceExpression",
        "value": ""
/*可选项, string, 人脸表情:“未知”, “无表情”, “高兴”, “惊喜”, “恐慌”, “悲伤”, “生气”, “傲慢”,
“厌烦” ;最大长度为 32 字节*/
    },
    {
        "description": "hat",
        "value": ""
/*可选项, string, 是否戴帽子:“no (否)”, “yes” (是) ;最大长度为 32 字节*/
    },
    {
        "description": "score",
        "value":
/*可选项, 人脸分数*/
    }
}],
    "contentID1": "",
/*人脸图片 ID*/
    "contentID2": "",
/*人脸背景图片 ID*/
    "FacePictureRect": {
/*可选项, 人脸缩略图帧信息*/
        "height": ,
/*必选项, float, 帧高度*/
```

```
        "width": ,
/*必选项, float, 帧宽度*/
        "x":/*必选项, float, 左上帧顶点 X 轴坐标*/
        "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
    }
    "pId1": "",
/*可选项, string, 人脸图片(包含头部和肩膀) ID 包含设备序列编号, 设备工作持续时间, 随机数字;最大长度为 32 字节*/
    "pId2": "",
/*可选项, string, 人脸背景图片 ID, 包含设备序列编号, 设备工作持续时间, 随机数字;最大长度为 32 字节*/
    "snapTime": ""
/*可选项, string, 人脸抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
    },
    "Human": {
/*可选项, 人体属性*/
        "Rect": {
/*可选项, 人体缩略图帧信息*/
            "height": ,
/*必选项, float, 帧高度*/
            "width": ,
/*必选项, float, 帧宽度*/
            "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
            "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
        },
        "Property": [{
/*人体属性详情*/
            "description": "",
            "value": ""
        }],
        "modeldata": "",
/*可选项, 人体模型数据(由 Base64 编码)*/
        "contentID1": "",
/*人体缩略图*/
        "contentID2": ""
/*人体背景图*/
        "pId1": "",
/*可选项, string, 人体或非机动车辆图片 ID, 最大长度为 32 字节*/
        "pId2": "",
/*可选项, string, 人体或非机动车辆背景图片 ID, 最大长度为 32 字节*/
        "snapTime": ""
/*可选项, string, 人体抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
    },
    "NonMotor": {
/*可选项, 非机动车辆属性*/
        "Rect": {
/*可选项, 非机动车辆缩略图帧信息*/
```

```
        "height": ,
/*必选项, float, 帧高度*/
        "width": ,
/*必选项, float, 帧宽度*/
        "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
        "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
    },
    "Property": [{
/*非机动车辆属性详情*/
        "description": "",
        "value": ""
    }],
    "contentID1": "",
/*可选项, string, 非机动车辆图片 ID*/
    "contentID2": "",
/*可选项, string, 非机动车辆背景图片 ID*/
    "pId1": "",
/*可选项, string, 非机动车辆图片 ID, 最大长度为 32 字节*/
    "pId2": "",
/*可选项, string, 非机动车辆背景图片 ID, 最大长度为 32 字节*/
    "snapTime": ""
/*可选项, string, 非机动车辆抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
    },
    "Vehicle": {
/*可选项, 车辆属性*/
        "VehicleRect": {
/*可选项, 车辆图片帧信息*/
            "height": ,
/*必选项, float, 帧高度*/
            "width": ,
/*必选项, float, 帧宽度*/
            "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
            "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
            "snapTime": ""
/*可选项, string, 车辆抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
        },
        "PlateRect": {
/*可选项, 车牌图片帧信息*/
            "height": ,
/*必选项, float, 帧高度*/
            "width": ,
/*必选项, float, 帧宽度*/
            "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
            "y":
```

```
/*必选项, float, 左上帧顶点 Y 轴坐标*/
    },
    "Property": [{
/*车辆属性详情*/
        "description": "",
        "value": ""
    }],
    "contentID1": "",
/*可选项, string, 车辆照片 ID*/
    "contentID2": "",
/*可选项, string, 车牌背景图片 ID*/
    "contentID3": ""
/*可选项, string, 车辆背景图片 ID*/
    "pId1": "",
/*可选项, string, 车辆图片 ID, 最大长度为 32 字节*/
    "pId2": "",
/*可选项, string, 车牌背景图片 ID, 最大长度为 32 字节*/
    "pId3": "",
/*可选项, string, 车辆背景图片 ID, 最大长度为 32 字节*/
    },
    "AbsoluteHigh":{
/*绝对云台位置**/
        "elevation": ,
/*可选项, int, 俯仰角, 取值范围: -900 到 2700*/
        "azimuth": ,
/*可选项, int, 偏转角, 取值范围: 0 到 3600*/
        "absoluteZoom":
/*可选项, int, 缩放率, 取值范围: 1 到 1000*/
    },
    "GPS":{
/*可选项, 设备经纬度信息*/
        "longitude": ,
/*必选项, float, 经度信息, 精确到小数点后 6 位, 取值范围为[-180.000000,180.000000]*/
        "latitude":
/*必选项, float, 纬度信息, 精确到小数点后 6 位, 取值范围为[-180.000000,180.000000]*/
    }
    "FaceContrastResult": [{
/*array, 人脸图片比对结果, 一个结果可以包含多人比对结果, 采用比对相似度最高的抓拍图片*/
        "errorCode": ,
/*可选项, int, 人脸图片分析失败时, 错误代码返回*/
        "errorMsg": "",
/*可选项, string, 人脸图片分析失败时, 错误信息返回*/
        "modelData": "",
/*可选项, string, 背景图片建模数据由 Base64 编码, 最大长度为 2048 字节*/
        "faces": [{
/*array, 人脸信息, 包括检测结果、性能、建模数据和匹配列表, 一个图片可以包含多个人脸*/
            "faceId": ,
/*可选项, integer32, 人脸 ID*/
            "identify": [{
```

```
/*可选项, 特性*/
    "relationId": "",
/*可选项, string, 布防联动 ID, 最大长度为 64 字节*/
    "maxsimilarity": ,
/*可选项, float, 匹配结果中最大相似性, 相似性为 0 到 1, 精确到小数点后 2 位*/
    "candidate": [{
/*array, 匹配的人脸照片信息*/
    "blacklist_id": "",
/*可选项, 人脸照片列表 ID 对应人脸库中 FDID, string, 最大长度为 64 字节*/
    "human_data": [{
/*array, 匹配人脸图片的人员信息*/
    "face_id": "",
/*可选项, string, 人脸 ID, 最大长度为 32 字节*/
    "contentID": "",
/*可选项, 人脸库图片 ID*/
    "similarity": ,
/*可选项, float, 人脸图片相似度从 0 到 1*/
    "isNoSaveFDPicture": ,
/*boolean, 是否保存人脸库图片: true (否), 如果该节点没有返回表示保存图片*/
    "pId": ""
/*可选项, string, 人脸库中图片 ID*/
    }],
    "human_id": "",
/*可选项, string, 人员 ID, string, 列表中人员 ID 对应人脸库中 FPID, 最大长度为 64 字节*/
    "reserve_field": {
/*可选项, 预留, 最大长度为 256 字节*/
    "name": "",
/*必选项, string, 人名, 最大长度为 96 字节*/
    "gender": "",
/*可选项, string, 性别: "male (男性)", "female (女性)", "unknown (未知)"; 最大长度为 10 字节*/
    "bornTime": "",
/*必选项, string, ISO 8601 格式出生日期, 最大长度为 32 字节*/
    "city": "",
/*可选项, string, 出生城市代码, 最大长度为 32 字节*/
    "certificateType": "",
/*可选项, string, 认证类型: "officerID", "ID" - 身份证, "passport (护照)", "other (其他)";
最大长度为 10 字节*/
    "certificateNumber": "",
/*可选项, string, 认证号码, 最大长度为 32 字节*/
    "ageGroup": "",
/*可选项, string, 年龄段: "unknown (未知)", "child (少年)", "young (青年)", "middle (中年)",
"old (老年)"; 最大长度为 32 字节*/
    "phoneNumber": ""
/*可选项, string, 电话号码, 最大长度为 64 字节*/
    "cityName": "石家庄市",
/*可选项, string, 城市名称 (与 city 字段对应), range:[0,32], desc:使用场景:客户端等计算能力较弱,不能实时根据 city 的编号查询出对应的城市名称*/
    "provinceName": "河北省"
/*可选项, string, 省份名称, range:[0,32], desc:使用场景:客户端等计算能力较弱,不能实时根据
```

```
city 的编号查询出对应的省份名称*/
    },
    "similarity": ,
/*可选项, float, 匹配结果中最高相似度值从 0 到 1, 精确到小数点后 2 位*/
    "listType": "",
/*可选项, string, 列表类型:“blacklist (黑名单)”, “whitelist (白名单)”, 最大长度为 20 字节*/
    "extendData": [{
/*可选项, 人员扩展信息*/
    "extendID": ,
/*可选项, int, 人员扩展信息 ID 从 1 开始*/
    "enable": ,
/*可选项, boolean, 是否启用人员信息扩展*/
    "name": "",
/*可选项, string, 人员标志扩展名, 最大长度为 96 字节*/
    "value": "",
/*可选项, string, 人员标志扩展内容, 最大长度为 64 字节*/
    }],
    "FDLibName": "",
/*可选项, 人脸库名称*/
    "FDLibThreshold": ,
/*可选项, 人脸库检测阈值, 取值范围为[0,100]*/
    "customFaceLibID": "",
/*可选项, string, 自定义人脸库 ID*/
    "customHumanID": ""
/*可选项, string, 自定义人员 ID*/
    }],
    "stayDuration":
/*可选项, integer32, 图像中巡航时间段, 单位: 毫秒*/
    }],
    "ModelingResult": {
/*可选项, 抓拍人脸图片建模结果*/
    "modelingStatus": "",
/*必选项, 建模状态*/
    "facePicQuality": "",
/*可选项, string, 人脸缩略图质量等级:“lower (低)”, “medium (中)”, “high (高)” */
    "stayDuration": ,
/*可选项, integer32, 图像中巡航时间段, 单位: 毫秒*/
    "modeldata": ""
/*可选项, 建模数据由 Base64 编码*/
    },
    "sid": "",
/*可选项, string, 场景 ID*/
    "sceneName": "",
/*可选项, string, 场景名称*/
    "uid": "",
/*事件 ID 可由时间 (精确到毫秒) 和任意数字组成, 最大长度为 64 字节*/
    "behaviorLinkageUid": "",
```



```
/*智能事件上传 ID 由时间（精确到毫秒）和任意数字组成，最大长度为 64 字节*/
    "position": ,
/*可选项，int，目标中心点偏移位置坐标，单位：米；例如，[500,100]*/
    "laneNo": ,
/*可选项，int，目标通道编号*/
    "speed": ,
/*可选项，float，目标速度*/
    "targetType": "",
/*可选项，string，“车辆”，“人”，“非车辆”*/
    "horizonSpeed": ,
/*可选项，目标水平速度，单位：米/秒*/
    "NormalizationPosition": ,
/*可选项，统一位置坐标，x 轴坐标值从 0 到 1000，y 轴坐标值从-500 到 1500，例如，[500,500]*/
    "targetMaxY": ,
/*可选项，float，目标最大 y 轴坐标值应小于 1500*/
    "TargetGPSInfo": [{
/*可选项，目标 GPS 信息*/
    "longitude": ,
/*必选项，float，经度，精确到小数点后 6 位，取值范围：[-180.000000,180.000000]*/
    "latitude":
/*必选项，float，纬度，精确到小数点后 6 位，取值范围：[-90.000000,90.000000]*/
    }],
    "RadarVideoTarget":{
/*可选项，当雷达或摄像机检测到目标时，目标属性会返回*/
    "radarDetected": ,
/*必选项，boolean，目标是否被雷达检测*/
    "position": ,
/*可选项，array of integer，相对于摄像机的目标中心偏移，单位：米。阵列中的元件代表 x 轴坐标和
y 轴坐标。对于 x 轴坐标，沿 x 轴向右为正，反向为负。对于 y 轴坐标，沿 y 轴向上为正，反向为负。摄像机
所在的水平线（与通道垂直）的中心点为原点，也是所有通道的中心点*/
    "speed": ,
/*可选项，float，目标速度*/
    "targetType": "",
/*可选项，string，目标类型：“vehicle”-机动车辆，“human（人）”，“nonMotor”-非机动车辆*/
    "vehicleParkingNum": ,
/*可选项，int，车辆停车次数*/
    "vehicleType": "",
/*可选项，string，车辆类型：“oversize”-大型车辆，“middle”-中型车辆，“light”-小型车辆*/
    "laneNo": ,
/*可选项，int，目标所在通道编号*/
    "horizonSpeed": ,
/*可选项，float，目标速度（水平速度），单位：米/秒*/
    "NormalizationPosition": ,
/*可选项，统一坐标，x 轴坐标取值范围：[0, 1000]，y 轴坐标取值范围：[-500, 1500]。这些值为统一
坐标后的初始坐标值，可能为负*/
    "targetMaxY": ,
/*可选项，float，目标 y 轴坐标的最大值。该值实际为坐标转换后 y 轴坐标最大值，最大值为 1500*/
    "licensePlate": "",
/*可选项，string，车牌号*/
```

```
"vehicleLogoRecog": ,
/*可选项, integer, 车辆主品牌*/
"vehileSubLogoRecog": ,
/*可选项, int, 车辆子品牌*/
"color":""
/*可选项, string, 车辆颜色*/
"MEFData": [{
/*可选项, array, 目标对应的移动电子围栏数据 (Mobile electronic fence), 可支持多个用户的电围
数据 (该数据需要进行敏感信息加密) */
"time": "2004-05-03T17:30:08+08:00.000",
/*必选项, datetime, 该电围数据采集时间, 精确到毫秒*/
"IMSI": "460001357924680",
/*必选项, string, SIM 卡号 (International Mobile Subscriber Identity), 最大长度为 15 字
节*/
"IMEI": "359355041886388",
/*可选项, string, 国际移动设备识别码 (International Mobile Equipment Identity), 最大长
度为 17 字节*/
"bandIdx": 11
/*可选项, int, 当前用户所使用的频带类型: 0 (联通 B3)、1 (电信 B3)、2 (移动 B38)、3 (移动 B39)、
4 (移动 B40)、5 (移动 B41)、6 (联通 B1)、7 (电信 B1)、8 (移动 2G)、9 (联通 2G)、10 (联通
WCDMA)、11 (电信 CDMA)、12 (电信 CDMA2000)、13 (移动 TD)、14 (WIFI 探针)、15 (联通 B5)、16
(电信 B5)、17 (联通 B8)、18 (电信 B8) */
}]
}
}],
"captureResultTargetNum": ,
/*可选项, int, 上传报警目标号码*/
}
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename="faceCapturePicture.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: faceImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" faceBackgroundImage.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: faceBackgroundImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" humanImage.jpg";
```

```
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: humanImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" humanBackgroundImage.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: humanBackgroundImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" nonMotorImage.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: nonMotorImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" vehicleImage.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: vehicleImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" plateImage.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: plateImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" vehicleBackgroundImage.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: vehicleBackgroundImage

fefefwageegfqaeg...
--MIME_boundary
```

```
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD1111";
filename=" faceLibImage.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: faceLibImage
```

```
fefefwageegfqaeg...
--MIME_boundary--
```

URL 格式图片上传

```
{
  "ipAddress": "",
  /*必选项, string, 报警设备 IPv4 地址, 最大长度为 32 字节, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备 IPv6 地址, 最大长度为 128 字节*/
  "portNo": ,
  /*可选项, integer32, 报警设备端口号*/
  "protocol": "",
  /*可选项, string, 协议类型:“HTTP”-设备网络 SDK,“HTTPS”,“EHome”-ISUP SDK;最大长度为 32 字节*/
  "macAddress": "",
  /*可选项, string, 物理地址, 最大长度为 32 字节, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
  /*可选项, integer32, 报警触发通道编号*/
  "dateTime": "",
  /*必选项, string, 报警触发时间 (ISO 8601 格式), 最大长度为 32 字节, 例如, 2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
  /*必选项, integer32, 一次报警上传时间*/
  "eventType": "",
  /*必选项, string, 已出现事件类型, 此处应设置为“mixedTargetDetection”;最大长度为 128 字节*/
  "eventState": "",
  /*必选项, string, 事件状态:“active-已出现”,“inactive”-未出现 (心跳数据);最大长度为 32 字节*/
  "eventDescription": "",
  /*必选项, 事件描述, 最大长度为 128 字节*/
  "channelName": "",
  /*必选项, string, 通道或摄像机名称*/
  "deviceID": "",
  /*可选项, string, 设备 ID, ISUP SDK 集成期间须返回*/
  "isDataRetransmission":,
  /*可选项, boolean, 数据重传标记*/
  "CaptureResult": [{
  /*抓拍和检测结果*/
    "targetID": ,
  /*必选项, int, 抓拍目标 ID*/
    "Face": {
  /*可选项, 人脸属性*/
      "Rect": {
```

```
/*可选项, 人脸缩略图帧信息*/
    "height": ,
/*必选项, float, 帧高度*/
    "width": ,
/*必选项, float, 帧宽度*/
    "x": ,
/*
/*必选项, float, 左上帧顶点 x 轴坐标*/
    "y":
/*必选项, float, 左上帧顶点 y 轴坐标*/
    },
    "Property": [{
/*人脸属性详情*/
        "description": "",
        "value": ""
    }
    {
        "description": "ageGroup",
        "value": ""
/*可选项, string, 年龄段:“unknown (未知)”, “child (少年)”, “young (青年)”, “middle (中年)”,
“old (老年)”;最大长度为 32 字节*/
    },
    {
        "description": "gender",
        "value": ""
/*可选项, string, 性别:“男性”, “女性”, “未知”;最大长度为 32 字节*/
    },
    {
        "description": "glass",
        "value": ""
/*可选项, string, 是否戴眼镜:“否”, “是”, “太阳镜”;最大长度为 32 字节*/
    },
    {
        "description": "smile",
        "value": ""
/*可选项, string, 是否微笑:“否”, “是”;最大长度为 32 字节*/
    },
    {
        "description": "mask",
        "value": ""
/*可选项, string, 是否戴口罩:“否”, “是”;最大长度为 32 字节*/
    },
    {
        "description": "faceExpression",
        "value": ""
/*可选项, string, 面部表情:“未知”, “无表情”, “高兴”, “惊喜”, “恐慌”, “悲伤”, “生气”, “傲慢”,
“厌烦”;最大长度为 32 字节*/
    },
    {
        "description": "hat",
```

```
"value": ""
/*可选项, string, 是否戴帽子:“否”,“是”,“未知”;最大长度为 32 字节*/
},
{
  "description": "score",
  "value":
/*可选项, 人脸分数*/
}],
  "FacePictureRect": {
/*可选项, 人脸缩略图帧信息*/
  "height": ,
/*必选项, float, 帧高度*/
  "width": ,
/*必选项, float, 帧宽度*/
  "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
  "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
}
  "faceImageURL": "",
/*可选项, 人脸图片 URL*/
  "faceBackgroundImageURL": ""
/*可选项, 人脸背景图片 URL*/
  "snapTime": ""
/*可选项, string, 人脸抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
},
  "Human": {
/*可选项, 人体属性*/
  "Rect": {
/*可选项, 人体缩略图帧信息*/
  "height": ,
/*必选项, float, 帧高度*/
  "width": ,
/*必选项, float, 帧宽度*/
  "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
  "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
},
  "Property": [{
/*人体属性详情*/
  "description": "",
  "value": ""
}],
  "modeldata": "",
/*可选项, 人体模型数据 (由 base64 编码) */
  "humanImageURL": "",
/*可选项, string, 人体或非机动车辆图片 URL*/
  "humanBackgroundImageURL": ""
/*可选项, string, 人体或非机动车辆背景图片 URL*/
```

```
    "snapTime": ""
/*可选项, string, 人体抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
  },
  "NonMotor": {
/*可选项, 非机动车辆属性*/
    "Rect": {
/*可选项, 非机动车辆缩略图帧信息*/
      "height": ,
/*必选项, float, 帧高度*/
      "width": ,
/*必选项, float, 帧宽度*/
      "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
      "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
    },
    "Property": [{
/*非机动车辆属性详情*/
      "description": "",
      "value": ""
    }],
    "nonMotorImageURL": "",
/*可选项, string, 非机动车辆图片 URL*/
    "nonMotorBackgroundImageURL": ""
/*可选项, string, 非机动车辆背景图片 URL*/
    "snapTime": ""
/*可选项, string, 非机动车辆抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
  },
  "Vehicle": {
/*可选项, 车辆属性*/
    "VehicleRect": {
/*可选项, 车辆图片帧信息*/
      "height": ,
/*必选项, float, 帧高度*/
      "width": ,
/*必选项, float, 帧宽度*/
      "x": ,
/*必选项, float, 左上帧顶点 X 轴坐标*/
      "y":
/*必选项, float, 左上帧顶点 Y 轴坐标*/
    }
    "snapTime": ""
/*可选项, string, 车辆抓拍时间, 最大长度为 32 字节, 例如, 2015-06-04T19:25:16-07:00*/
  },
  "PlateRect": {
/*可选项, 车牌照照片帧信息*/
    "height": ,
/*必选项, float, 帧高度*/
    "width": ,
/*必选项, float, 帧宽度*/
    "x": ,
```

```
/*必选项, float, 左上帧顶点 x 轴坐标*/
    "y":
/*必选项, float, 左上帧顶点 y 轴坐标*/
    },
    "Property": [{
/*车辆属性详情*/
        "description": "",
        "value": ""
    }],
    "vehicleImageURL": "",
/*可选项, string, 车辆图片 URL*/
    "plateImageURL": "",
/*可选项, string, 车牌图片 URL*/
    "vehicleBackgroundImageURL": ""
/*可选项, string, 车辆背景图片 URL*/
    },
    "AbsoluteHigh":{
/*绝对 PTZ 位置*/
        "elevation": ,
/*可选项, int, 俯仰角, 取值范围: -900 到 2700*/
        "azimuth": ,
/*可选项, int, 偏转角, 取值范围: 0 到 3600*/
        "absoluteZoom":
/*可选项, int, 缩放率, 取值范围: 1 到 1000*/
    },
    "GPS":{
/*可选项, 设备经纬度信息*/
        "longitude": ,
/*必选项, float, 经度信息, 精确到小数点后 6 位, 取值范围为[-180.000000,180.000000]*/
        "latitude":-90.000000
/*必选项, float, 纬度信息, 精确到小数点后 6 位, 取值范围为[-180.000000,180.000000]*/
    },
    "FaceContrastResult": [{
/*array, 人脸图片比对结果, 一个结果会包含多人比对结果, 采用比对相似度最高的抓拍图片*/
        "errorCode": ,
/*可选项, int, 分析人脸图片失败时, 错误代码返回*/
        "errorMsg": "",
/*可选项, string, 分析人脸图片失败时, 错误信息返回*/
        "modelData": "",
/*可选项, string, 背景图片建模数据由 Base64 编码, 最大长度为 2048 字节*/
        "faces": [{
/*array, 人脸信息, 包括检测结果、特性、建模数据和匹配列表, 一个图片可以包含多个人脸*/
            "faceId": ,
/*可选项, integer32, 人脸 ID*/
            "identify": [{
/*可选项, 特征*/
                "relationId": "",
/*可选项, string, 布防联动 ID, 最大长度为 64 字节*/
                "maxsimilarity": ,
```



```
/*可选项, float, 匹配结果中最大相似度从 0 到 1, 精确到小数点后 2 位*/
    "candidate": [{
/*array, 匹配的人脸图片信息*/
        "blacklist_id": "",
/*可选项, 人脸图片列表 ID, 人脸图片列表 ID 对应人脸库中 FDID, string, 最大长度为 64 字节*/
        "human_data": [{
/*array, 已匹配人脸图片的人员信息*/
            "face_id": "",
/*可选项, string, 人脸 ID, 最大长度为 32 字节*/
            "similarity": ,
/*可选项, float, 人脸图片相似度从 0 到 1*/
            "isNoSaveFDPicture": ,
/*boolean, 是否保存人脸库图片: true- (否), 如果该节点没有返回表示保存图片*/
            "face_picurl": "",
/*可选项, string, 人脸图片 URL*/
        }],
        "human_id": "",
/*可选项, string, 人员 ID, string, 列表中人员 ID 对应人脸库中 FDID, 最大长度为 64 字节*/
        "reserve_field": {
/*可选项, 预留, 最大长度为 256 字节*/
            "name": "",
/*必选项, string, 人名, 最大长度为 96 字节*/
            "gender": "",
/*可选项, string, 性别: "male (男性)", "female (女性)", "unknown (未知)"; 最大长度为 10 字节*/
            "bornTime": "",
/*必选项, string, ISO 8601 出生日期, 最大长度为 32 字节*/
            "city": "",
/*可选项, string, 出生城市代码, 最大长度为 32 字节*/
            "certificateType ": "",
/*可选项, string, 认证类型: "officerID", "ID" -身份证, "passport (护照)", "other (其他)";
最大长度为 10 字节*/
            "certificateNumber": "",
/*可选项, string, 证书编号, 最大长度为 32 字节*/
            "ageGroup": "",
/*可选项, string, 年龄段: "unknown (未知)", "child (少年)", "young (青年)", "middle (中年)",
"old (老年)"; 最大长度为 32 字节*/
            "phoneNumber": "",
/*可选项, string, 电话号码, 最大长度为 64 字节*/
            "cityName": "石家庄市",
/*可选项, string, 城市名称 (与 city 字段对应), 长度范围: [0, 32]。使用场景: 客户端等计算能力较
弱, 不能实时根据 city 的编号查询出对应的城市名称*/
            "provinceName": "河北省"
/*可选项, string, 省份名称, 长度范围: [0, 32]。使用场景: 客户端等计算能力较弱, 不能实时根据 city
的编号查询出对应的省份名称*/
        }],
        "similarity": ,
/*可选项, float, 匹配结果中最高相似度值从 0 到 1, 精确到小数点后 2 位*/
        "listType": "",
/*可选项, string, 列表类型: "黑名单", "白名单", 最大长度为 20 字节*/
```

```

        "extendData": [{
/*可选项，人员扩展信息*/
            "extendID": ,
/*可选项，int，人员扩展信息 ID 从 1 开始*/
            "enable": ,
/*可选项，boolean，是否启用人员信息扩展*/
            "name": "",
/*可选项，string，人员标签扩展名称，最大长度为 96 字节*/
            "value": "",
/*可选项，string，人员标签扩展内容，最大长度为 64 字节*/
        }],
        "FDLibName": "",
/*可选项，人脸库名称*/
        "FDLibThreshold": ,
/*可选项，人脸库检测阈值，取值范围为[0,100]*/
        "customFaceLibID": "",
/*可选项，string，自定义人脸库 ID*/
        "customHumanID": ""
/*可选项，string，自定义人员 ID*/
    }],
    "stayDuration":
/*可选项，integer32，图像中巡航时间段，单位：毫秒*/
    },
    ],
    "ModelingResult": {
/*可选项，抓拍人脸图片建模结果*/
        "modelingStatus": "",
/*必选项，建模状态*/
        "facePicQuality": "",
/*可选项，string，人脸缩略图质量等级：“lower（低）”，“medium（中）”，“high（高）” */
        "stayDuration": ,
/*可选项，integer32，图像中巡航时间段，单位：毫秒*/
        "modeldata": ""
/*可选项，建模数据由 Base64 编码*/
    },
    "sid": "",
/*可选项，string，场景 ID*/
    "sceneName": "",
/*可选项，string，场景名称*/
    "uid": "",
/*事件 ID 可由时间（精确到毫秒）和任意数字组成，最大长度为 64 字节*/
    "behaviorLinkageUid": "",
/*智能事件上传 ID 可由时间（精确到毫秒）和任意数字组成，最大长度为 64 字节*/
    "position": ,
/*可选项，int，目标中心点偏移位置坐标，单位：米，例如，[500,100]*/
    "laneNo": ,
/*可选项，int，目标通道编号*/
    "speed": ,

```

```
/*可选项, float, 目标速度*/
    "targetType": "",
/*可选项, string, “车辆”, “人”, “非机动车辆” */
    "horizonSpeed": ,
/*可选项, 目标水平速度, 单位: 米/秒*/
    "NormalizationPosition": ,
/*可选项, 统一位置坐标, x 轴坐标取值为 0 到 1000, y 轴坐标取值为-500 到 1500, 例如, [500,500]*/
    "targetMaxY": ,
/*可选项, float, 目标最大 Y 轴坐标值应小于 1500*/
    "TargetGPSInfo": [{
/*可选项, 目标 GPS 信息*/
        "longitude": ,
/*必选项, float, 经度, 精确到小数点后 6 位, 取值范围: [-180.000000,180.000000]*/
        "latitude":
/*必选项, float, 纬度, 精确到小数点后 6 位, 取值范围: [-90.000000,90.000000]*/
    }],
    "perimeterLinkageUid"
/*可选项, string, 已绑定的周界事件上传唯一 ID, 最大长度为 64 字节*/
    "RadarVideoTarget":{
/*可选项, 当雷达或摄像机探测目标时, 雷达目标属性会返回*/
        "radarDetected": ,
/*必选项, boolean, 雷达探测是否有效*/
        "position": ,
/*可选项, array of integer, 相对于摄像机的目标中心偏移, 单位: 米。阵列中的元件代表 x 轴坐标和
y 轴坐标。对于 x 轴坐标, 沿 x 轴向右为正, 反向为负。对于 y 轴坐标, 沿 y 轴向上为正, 反向为负。摄像机
所在的水平线 (与通道垂直) 的中心点为原点, 也是所有通道的中心点*/
        "speed": ,
/*可选项, float, 目标速度*/
        "targetType": "",
/*可选项, string, 目标类型: “vehicle” -机动车辆, “human (人)”, “nonvehicle” -非机动车辆*/
        "vehicleParkingNum": ,
/*可选项, int, 车辆停车时间*/
        "vehicleType": "",
/*optional, string, vehicle type: “oversize” -大型车辆, “middle” -中型车辆, “light” -
小型车辆*/
        "laneNo": ,
/*可选项, int, 目标所在车道编号*/
        "horizonSpeed": ,
/*可选项, float, 目标速度 (水平速度), 单位: 米/秒*/
        "NormalizationPosition": ,
/*可选项, 统一坐标, x 轴坐标取值范围: [0, 1000], y 轴坐标取值范围: [-500, 1500]。这些值为统一
坐标后的初始坐标值, 可能为负*/
        "targetMaxY": ,
/*可选项, float, 目标 Y 轴坐标的最大值。该值实际为坐标转换后 Y 轴坐标最大值, 最大值为 1500*/
        "licensePlate": "",
/*可选项, string, 车牌号*/
        "vehicleLogoRecog": ,
/*可选项, int, 车辆主品牌*/
        "vehileSubLogoRecog": ,
```

```

/*可选项, int, 车辆子品牌*/
    "color":""
/*可选项, string, 车辆颜色*/
    },
    "MEFData": [{
/*可选项, array, 目标对应的移动电子围栏数据 (Mobile electronic fence), 可支持多个用户的电围
数据 (该数据需要进行敏感信息加密) */
        "time": "2004-05-03T17:30:08+08:00.000",
/*必选项, datetime, 该电围数据采集时间, 精确到毫秒*/
        "IMSI": "460001357924680",
/*必选项, string, SIM 卡号 (International Mobile Subscriber Identity), 最大长度为 15 字
节*/
        "IMEI": "359355041886388",
/*可选项, string, 国际移动设备识别码 (International Mobile Equipment Identity), 最大长
度为 17 字节*/
        "bandIdx": 11
/*可选项, int, 当前用户所使用的频带类型: 0 (联通 B3)、1 (电信 B3)、2 (移动 B38)、3 (移动 B39)、
4 (移动 B40)、5 (移动 B41)、6 (联通 B1)、7 (电信 B1)、8 (移动 2G)、9 (联通 2G)、10 (联通
WCDMA)、11 (电信 CDMA)、12 (电信 CDMA2000)、13 (移动 TD)、14 (WIFI 探针)、15 (联通 B5)、16
(电信 B5)、17 (联通 B8)、18 (电信 B8) */
    }]
}],
    "captureResultTargetNum":
/*可选项, int, 上传的报警目标编号*/
    "triggerSourceType"
/*可选项, string, 触发设备类型: 如果该节点没有返回或返回的值为 "ruleTrigger" -已配置的混合目
标检测规则触发报警 (相关 URI: /ISAPI/Intelligent/channels/<ID>/mixedTargetDetection?
format=json), "linkedTrackingCapture" -联动跟踪抓拍 (相关 URI: /ISAPI/
MasterSlaveTracking/linkedTracking?format=json) */
}

```

16.4.10 JSON_EventNotificationAlert_PeopleQueuingUpAlarmMsg

关于排队人数检测报警细节的 JSON 报文

排队人数检测报警包含二进制图片数据

```

Content-Type: multipart/form-data; boundary=MIME_boundary
--MIME_boundary
Content-Type: application/json
Content-Length: 480
{
  "ipAddress": "",
  "ipv6Address": "",
  "portNo": ,
  "protocol": "",
  "macAddress": "",
  "channelID": "",

```

```
"releatedChannelList": [1,2,3],
/*可选项, array of integers, 报警关联（同源）通道列表（与 channelID 在同一个相机上的通道），
用于平台收到报警时展示预览或回放*/
  "dateTime": "",
  "activePostCount": ,
  "eventType": "",
/*必选项, string, 触发的事件类型："personQueueCounting"-排队人数检测，最大长度为 128*/
  "eventState": "",
  "eventDescription": "",
  "deviceID": "",
  "RegionCapture"{
/*区域抓拍结果信息*/
    "humanCounting":{
/*人数统计*/
      "count":
/*必选项, int, 规则区域统计的实际人数值，范围[0,60]*/
    }
    "rule":{
/*规则属性*/
      "ruleID": ,
/*必选项, int, 该人员对应的规则区域 ID，与配置中的规则 ID*/
      "alarmCount": ,
/*必选项, int, 规则区域统计人数报警值，范围：[0,60]*/
      "regionColor": "",
/*必选项, string, 规则区域颜色："blue"-蓝, "red"-红, "yellow"-黄*/
      "Region": [{
/*必选项, 规则区域坐标, 3-10 边形*/
        "x": ,
/*必选项, float, x 坐标, 范围：[0.000,1]*/
        "y":
/*必选项, float, y 坐标, 范围：[0.000,1]*/
      }]
      "countTriggerType": "",
/*可选项, string, 排队人数检测报警触发类型："greater"-排队人数大于 alarmCount 值，报警触发,
"less"-排队人数小于 alarmCount 值，报警触发, "equal"-排队人数等于 alarmCount 值，报警触发,
"unequal"-排队人数不等于 alarmCount 值，报警触发, "range"-排队人数在 maxAlarmCount 和
alarmCount 的值之间，报警触发*/
      "durationTime":
/*可选项, int, 排队人数报警持续时间；范围：[0,3600]s, 默认为 300s*/
    }
  }
  "contentID": ""
/*可选项, string, 背景图，最大长度为 64*/

}
--MIME_boundary
Content-Type: image/jpeg
Content-Length: 42569
Content-ID: human_image
```

```
fefefwageegfqaeg...
--MIME_boundary--
```

排队人数检测报警包含图片 URL

```
{
  "ipAddress": "",
  "ipv6Address": "",
  "portNo": ,
  "protocol": "",
  "macAddress": "",
  "channelID": "",
  "relatedChannelList": [1,2,3],
  /*可选项, array of integers, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道),
  用于平台收到报警时展示预览或回放*/
  "dateTime": "",
  "activePostCount": ,
  "eventType": "",
  /*必选项, string, 触发的事件类型: "personQueueCounting"-排队人数检测, 最大长度为 128*/
  "eventState": "",
  "eventDescription": "",
  "deviceID": "",
  "RegionCapture"{
  /*区域抓拍结果信息*/
    "humanCounting":{
  /*人数统计*/
      "count":
  /*必选项, int, 规则区域统计的实际人数值, 范围[0,60]*/
    }
    "rule":{
  /*规则属性*/
      "ruleID": ,
  /*必选项, int, 该人员对应的规则区域 ID, 与配置中的规则 ID*/
      "alarmCount": ,
  /*必选项, int, 规则区域统计人数报警值, 范围: [0,60]*/
      "regionColor": "",
  /*必选项, string, 规则区域颜色: "blue"-蓝, "red"-红, "yellow"-黄*/
      "Region": [{
  /*必选项, 规则区域坐标, 3-10 边形*/
        "x": ,
  /*必选项, float, x 坐标, 范围: [0.000,1]*/
        "y":
  /*必选项, float, y 坐标, 范围: [0.000,1]*/
      }]
      "countTriggerType": "",
  /*可选项, string, 排队人数检测报警触发类型: "greater"-排队人数大于 alarmCount 值, 报警触发,
  "less"-排队人数小于 alarmCount 值, 报警触发, "equal"-排队人数等于 alarmCount 值, 报警触发,
  "unequal"-排队人数不等于 alarmCount 值, 报警触发, "range"-排队人数在 maxAlarmCount 和
  alarmCount 的值之间, 报警触发*/
      "durationTime":
```

```

/*可选项, int, 排队人数报警持续时间;范围:[0,3600]s, 默认为 300s*/
}
}
"backgroundImageURL": ""
/*可选项, string, 背景图 URL, 最大长度为 256*/
}

```

16.4.11 JSON_EventNotificationAlert_UnregisteredStreetVendorMsg

关于非法摆摊事件详情的 JSON 报文

图片以二进制形式上传

```

{
  "ipAddress": "",
  /*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 位, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 位*/
  "portNo": ,
  /*可选项, integer32, 报警设备的端口号*/
  "protocol": "",
  /*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
  传输;其最大取值长度为 32 位*/
  "macAddress": "",
  /*可选项, string, MAC 地址, 其最大取值长度为 32 位, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
  /*可选项, integer32, 报警触发通道号*/
  "dateTime": "",
  /*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 位, 例如,
  2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
  /*必选项, integer32, 单个报警上传次数*/
  "eventType": "",
  /*必选项, string, 事件类型, 此处需要设置为“cityManagement”*/
  "eventState": "",
  /*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生 (心跳数据);其最大取值长度
  为 32 位*/
  "eventDescription": "",
  /*必选项, 事件描述, 其最大取值长度为 128 位*/
  "deviceID": "",
  /*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
  "Result": [{
    "ruleID": ,
    /*必选项, int, 规则 ID*/
    "subEventType": "",
    /*必选项, int, 事件子类型, 此处需设置为“UnregisteredStreetVendor”*/
    "Target": [{
    /*必选项, 目标信息, 只有一个目标可上传*/

```

```

    "targetID": ,
/*可选项，目标 ID*/
    "Rect": {
/*可选项，目标矩形框*/
    "height": ,
/*必选项，float，矩形框的高度*/
    "width": ,
/*必选项，float，矩形框的高度*/
    "x": ,
/*必选项，float，矩形框左上角的 x 坐标*/
    "y":
/*必选项，float，矩形框左上角的 y 坐标*/
    },
    "contentID": "",
/*目标图片 ID*/
    },
    "contentID": "",
/*目标背景图片 ID*/
    "AbsoluteHigh": {
/*可选项，PTZ 位置*/
    "elevation": ,
/*可选项，int，上下角度，其取值范围从-900 到 2700*/
    "azimuth": ,
/*可选项，int，左右角度，其取值范围从 0 到 3600*/
    "absoluteZoom":
/*可选项，int，缩放倍率，其取值范围从 1 到 1000*/
    }
    }
}

```

图片以 URL 形式上传

```

{
    "ipAddress": "",
/*必选项，string，报警设备的 IPv4 地址，其最大取值长度为 32 位，例如，172.6.64.7*/
    "ipv6Address": "",
/*必选项，string，报警设备的 IPv6 地址，其最大取值长度为 128 位*/
    "portNo": ,
/*可选项，integer32，报警设备的端口号*/
    "protocol": "",
/*可选项，string，协议类型：“HTTP”-用于设备网络 SDK 传输，“HTTPS”，“EHome”-用于 ISUP SDK
传输；其最大取值长度为 32 位*/
    "macAddress": "",
/*可选项，string，MAC 地址，其最大取值长度为 32 位，例如，01:17:24:45:D9:F4*/
    "channelID": ,
/*可选项，integer32，报警触发通道号*/
    "dateTime": "",
/*必选项，string，报警触发时间（ISO 8601 格式），其最大取值长度为 32 位，例如，
2004-05-03T17:30:08+08:00*/

```



```

    "activePostCount": ,
    /*必选项, integer32, 单个报警上传次数*/
    "eventType": "",
    /*必选项, string, 事件类型, 此处需要设置为“cityManagement” */
    "eventState": "",
    /*必选项, string, 事件状态:“active”-已发生, “inactive”-未发生(心跳数据);其最大取值长度为32位*/
    "eventDescription": "",
    /*必选项, 事件描述, 其最大取值长度为128位*/
    "deviceID": "",
    /*可选项, string, 设备ID, 在ISUP SDK传输过程中, 该节点必选*/
    "Result": [{
        "ruleID": ,
        /*必选项, int, 规则ID*/
        "subEventType": "",
        /*必选项, int, 事件子类型, 此处需设置为“UnregisteredStreetVendor” */
        "Target": [{
            /*必选项, 目标信息, 只有一个目标可上传*/
            "targetID": ,
            /*可选项, 目标ID*/
            "Rect": {
                /*可选项, 目标矩形框*/
                "height": ,
                /*必选项, float, 矩形框的高度*/
                "width": ,
                /*必选项, float, 矩形框的高度*/
                "x": ,
                /*必选项, float, 矩形框左上角的x坐标*/
                "y": ,
                /*必选项, float, 矩形框左上角的y坐标*/
            },
            "contentID": "",
            /*目标图片ID*/
        }],
        "targetImageURL": "",
        /*目标图片URL*/
    },
    "backgroundImageURL": ""
    /*目标背景图片URL*/
    }]
}

```

16.4.12 JSON_EventNotificationAlert_WaitingTimeDetectionMsg

关于区域等待时长检测事件详情的JSON报文

图片以二进制形式上传

```
Content-Type: multipart/form-data; boundary=MIME_boundary
--MIME_boundary
Content-Type: application/json
Content-Length: 480
{
  "ipAddress": "",
  /*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 位, 例如, 172.6.64.7*/
  "ipv6Address": "",
  /*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 位*/
  "portNo": ,
  /*可选项, integer32, 报警设备的端口号*/
  "protocol": "",
  /*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
  传输;其最大取值长度为 32 位*/
  "macAddress": "",
  /*可选项, string, MAC 地址, 其最大取值长度为 32 位, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
  /*可选项, integer32, 报警触发通道号*/
  "dateTime": "",
  /*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 位, 例如,
  2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
  /*必选项, integer32, 单个报警上传次数*/
  "eventType": "",
  /*必选项, string, 事件类型, 此处需要设置为“personQueueTime”*/
  "eventState": "",
  /*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生 (心跳数据);其最大取值长度
  为 32 位*/
  "eventDescription": "",
  /*必选项, 事件描述, 其最大取值长度为 128 位*/
  "deviceID": "",
  /*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
  "humanCapture": {
  /*抓拍的人体信息*/
    "human":{
  /*人员属性*/
      "targetID": ,
  /*必选项, int, 人员目标 ID, 该目标等待时长比设置的报警阈值长, 其 ID 由 DSP 生成, 和人员位置对应*/
      "waitTime": ,
  /*必选项, int, 实际等待时长, 其取值范围从 0 到 3600, 单位: 秒*/
      "Region": [{
  /*必选项, 目标人员区域坐标, 区域为四边形*/
        "x": ,
  /*必选项, float, 区域 X 坐标, 取值范围从 0.000 到 1*/
        "y":
  /*必选项, float, 区域 Y 坐标, 取值范围从 0.000 到 1*/
      }]
    }
  }
```

```

        contentID": ""
/*抓拍的人体图片*/
    },
    "rule":{
/*规则属性*/
        "ruleID": ,
/*必选项, int, 规则 ID, 与目标人员区域 ID 对应*/
        "alarmTime": ,
/*required, int, 配置的等待时长报警阈值, 取值范围从 0 到 3600, 单位: 秒*/
        "regionColor": "",
/*必选项, string, 规则区域颜色: "blue" -蓝色, "red" -红色, "yellow" -黄色*/
        "Region": [{
/*必选项, 多边形规则区域坐标 (包含 3 到 10 条边) */
            "x": ,
/*必选项, float, x 坐标, 其取值范围从 0.000 到 1*/
            "y":
/*必选项, float, x 坐标, 其取值范围从 0.000 到 1*/
        }]
        "timeTriggerType": ""
/*可选项, string, 等待时长报警的触发类型: "greater" -高于报警阈值, "less" -低于报警阈值,
"equal" -等于报警阈值, "unequal" -不等于报警阈值*/
    }
},
"contentID": ""
/*背景图片*/
}
--MIME_boundary
Content-Type: image/jpeg
Content-Length: 42569
Content-ID: human_image

fefefwageegfqaeg...
--MIME_boundary
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: background_image

fefefwageegfqaeg...
--MIME_boundary--

```

图片以 URL 形式上传

```

{
    "ipAddress": "",
/*必选项, string, 报警设备的 IPv4 地址, 其最大取值长度为 32 位, 例如, 172.6.64.7*/
    "ipv6Address": "",
/*必选项, string, 报警设备的 IPv6 地址, 其最大取值长度为 128 位*/
    "portNo": ,
/*可选项, integer32, 报警设备的端口号*/
    "protocol": "",

```

```
/*可选项, string, 协议类型:“HTTP”-用于设备网络 SDK 传输,“HTTPS”,“EHome”-用于 ISUP SDK
传输;其最大取值长度为 32 位*/
  "macAddress": "",
/*可选项, string, MAC 地址, 其最大取值长度为 32 位, 例如, 01:17:24:45:D9:F4*/
  "channelID": ,
/*可选项, integer32, 报警触发通道号*/
  "dateTime": "",
/*必选项, string, 报警触发时间 (ISO 8601 格式), 其最大取值长度为 32 位, 例如,
2004-05-03T17:30:08+08:00*/
  "activePostCount": ,
/*必选项, integer32, 单个报警上传次数*/
  "eventType": "",
/*必选项, string, 事件类型, 此处需要设置为“personQueueTime”*/
  "eventState": "",
/*必选项, string, 事件状态:“active”-已发生,“inactive”-未发生 (心跳数据);其最大取值长度
为 32 位*/
  "eventDescription": "",
/*必选项, 事件描述, 其最大取值长度为 128 位*/
  "deviceID": "",
/*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
  "humanCapture": {
/*抓拍的人体信息*/
  "human":{
/*人员属性*/
  "targetID": ,
/*必选项, int, 人员目标 ID, 该目标等待时长比设置的报警阈值长, 其 ID 由 DSP 生成, 和人员位置对应*/
  "waitTime": ,
/*必选项, int, 实际等待时长, 其取值范围从 0 到 3600, 单位:秒*/
  "Region": [{
/*必选项, 目标人员区域坐标, 区域为四边形*/
  "x": ,
/*必选项, float, 区域 X 坐标, 取值范围从 0.000 到 1*/
  "y":
/*必选项, float, 区域 Y 坐标, 取值范围从 0.000 到 1*/
  }]}
  "humanImageURL": ""
/*抓拍的人体图片 URL*/
  },
  "rule":{
/*规则属性*/
  "ruleID": ,
/*必选项, int, 规则 ID, 与目标人员区域 ID 对应*/
  "alarmTime": ,
/*required, int, 配置的等待时长报警阈值, 取值范围从 0 到 3600, 单位:秒*/
  "regionColor": "",
/*必选项, string, 规则区域颜色:“blue”-蓝色,“red”-红色,“yellow”-黄色*/
  "Region": [{
/*必选项, 多边形规则区域坐标 (包含 3 到 10 条边) */
  "x": ,
```

```

/*必选项, float, X 坐标, 其取值范围从 0.000 到 1*/
    "y":
/*必选项, float, X 坐标, 其取值范围从 0.000 到 1*/
    }]
    "timeTriggerType": ""
/*可选项, string, 等待时长报警的触发类型:“greater”-高于报警阈值,“less”-低于报警阈值,
“equal”-等于报警阈值,“unequal”-不等于报警阈值*/
    }
},
"backgroundImageURL": ""
/*背景图片 URL*/
}

```

16.4.13 JSON_ResponseStatus

关于响应状态的 JSON 报文

```

{
    "requestURL": "",
/*可选项, string, 请求 URL*/
    "statusCode": ,
/*可选项, int, 状态码*/
    "statusString": "",
/*可选项, string, 状态描述*/
    "subStatusCode": "",
/*可选项, string, 子状态码*/
    "errorCode": ,
/*必选项, int, 错误码, 与 subStatusCode 对应, statusCode 不为 1 时, 该字段为必选项, 返回值转换
后将带小数*/
    "errorMsg": "",
/*必选项, string, 错误信息, statusCode 不为 1 时, 该字段为必选项*/
    "MErrCode": "0xFFFFFFFF",
/*可选项, string, 按功能模块分类的错误码*/
    "MErrDevSelfEx": "0xFFFFFFFF"
/*可选项, string, MErrCode 的扩展字段, 用于定义自定义错误码, 按照功能模块分类*/
}

```

说明

- 关于状态码、子状态码、错误码和错误描述的详情, 参见 [ISAPI 返回码](#)。
- 关于错误码、错误描述和调试建议的详情, 参见 [功能模块化错误码](#)。

16.4.14 XML_DeviceCap

关于设备能力的 XML 报文

```

<DeviceCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <SysCap><!--可选项-->
    <isSupportDst><!--可选项, xs: boolean, 是否支持夏令时--></isSupportDst>
    <NetworkCap/><!--可选项, xs: boolean, 网络能力-->
    <IOCap/><!--可选项, 报警输入输出能力-->
    <SerialCap/><!--可选项, 串口能力-->
    <VideoCap/><!--可选项, 视频能力, 可参见报文 XML_VideoCap-->
    <AudioCap/><!--可选项, 音频能力-->
    <isSupportHoliday><!--可选项, xs:boolean--></isSupportHoliday>
    <RebootConfigurationCap>
      <Genetec><!--可选项, xs:boolean--></Genetec>
      <ONVIF><!--可选项, xs:boolean--></ONVIF>
      <RTSP><!--可选项, xs:boolean--></RTSP>
      <HTTP><!--可选项, xs:boolean--></HTTP>
      <SADP>
        <ISDiscoveryMode><!--可选项, xs:boolean--></ISDiscoveryMode>
        <PcapMode><!--可选项, xs:boolean--></PcapMode>
      </SADP>
      <IPCAddStatus><!--可选项, xs:boolean--></IPCAddStatus>
    </RebootConfigurationCap>
    <isSupportExternalDevice><!--可选项, xs:boolean, 设备是否支持接入外设: true-支持,
    不支持时不返回该节点, 对应的 URI 为/ISAPI/System/externalDevice/capabilities--></
    isSupportExternalDevice>
    <isSupportSoundCfg><!--可选项, xs:boolean, 是否支持声音配置, 对应的 URI 为/ISAPI/
    System/SoundCfg/capabilities--></isSupportSoundCfg>
    <isSupportSubscribeEvent><!--可选项, xs:boolean, 是否支持事件订阅, 对应的 URI 为/
    ISAPI/Event/notification/subscribeEventCap--></isSupportSubscribeEvent>
    <isSupportSubscribeIOTInfo><!--可选项, xs:boolean, 是否支持物联网信息订阅--></
    isSupportSubscribeIOTInfo>
    <isSupportDiagnosedData><!--可选项, xs:boolean, 是否支持设备诊断数据导出功能(true-
    支持, 不支持不返回该节点), 对应的 URI (/ISAPI/System/diagnosedData)--></
    isSupportDiagnosedData>
    <isSupportChangedUpload>
      <!--可选项, xs: boolean, 是否支持状态变化上报-->
    </isSupportChangedUpload>
    <isSupportGettingWorkingStatus>
      <!--可选项, xs:boolean, 是否支持获取设备状态-->
    </isSupportGettingWorkingStatus>
    <isSupportGettingChannelInfoByCondition>
      <!--可选项, xs:boolean-->
    </isSupportGettingChannelInfoByCondition>
    <isSupportDiagnosedDataParameter>
      <!--可选项, xs:boolean-->
    </isSupportDiagnosedDataParameter>
    <isSupportSimpleDevStatus>
      <!--可选项, xs: boolean, 是否支持获取设备工作状态-->
    </isSupportSimpleDevStatus>
    <isSupportFlexible>

```

```

    <!--可选项, xs:boolean, 是否支持根据条件获取通道状态-->
</isSupportFlexible>
<isSupportPTZChannels>
    <!--可选项, xs:boolean, 是否支持返回 PTZ 通道 (区分视频通道) -->
</isSupportPTZChannels>
<isSupportDiagnosedData>
    <!--可选项, xs:boolean, 是否支持数据诊断-->
</isSupportDiagnosedData>
<isSupportTimeCap>
    <!--可选项, xs:boolean, 是否支持时间能力-->
</isSupportTimeCap>
<isSupportThermalStreamData>
    <!--可选项, xs:boolean, 是否支持热成像码流数据实时上传, 若支持, 则返回该节点且值为
"true"; 若不支持, 则不返回该节点-->
</isSupportThermalStreamData>
<isSupportPostUpdateFirmware>
    <!--可选项, xs:boolean, 是否支持 POST 方法固件升级-->
</isSupportPostUpdateFirmware>
<isSupportPostConfigData>
    <!--可选项, xs:boolean, 是否支持导入或导出配置文件-->
</isSupportPostConfigData>
<isSupportUserLock>
    <!--可选项, xs:boolean, 是否支持用户锁定-->
</isSupportUserLock>
<isSupportModuleLock><!--可选项, xs:boolean, 是否支持模块锁定--></
isSupportModuleLock>
<isSupportSoundCfg><!--可选项, xs:boolean--></isSupportSoundCfg>
<isSupportMetadata>
    <!--可选项, xs:boolean, 若支持, 则返回且值为"true", 否则不返回该节点-->
</isSupportMetadata>
<isSupportShutdown><!--可选项, xs:boolean, 是否支持关机配置--></isSupportShutdown>
<isSupportConsumptionMode><!--可选项, xs:boolean, 设备是否支持功耗模式切换: true-支持--></isSupportConsumptionMode>
<isSupportManualPowerConsumption><!--可选项, xs:boolean, 设备是否支持功耗手动控制:
true-支持, 不支持不返回该节点--></isSupportManualPowerConsumption>
<supportSmartOverlapChannles opt="1"/><!--可选项, xs:boolean, 是否支持智能事件的码
流配置, 若支持, 则返回该节点和对应通道 ID, 否则不返回该节点-->
</SysCap>
<voicetalkNums><!--可选项, xs:integer, 语音对讲通道数量--></voicetalkNums>
<isSupportSnapshot><!--可选项, xs:boolean, 是否支持抓图--></isSupportSnapshot>
<SecurityCap/><!--可选项, 安全能力-->
<EventCap/><!--可选项, 事件能力 (相关 URI: /ISAPI/Event/capabilities) -->
<ITCCap><!--可选项--></ITCCap>
<ImageCap/><!--可选项, 图片能力-->
<RacmCap/><!--可选项, 存储能力-->
<PTZCtrlCap>
    <isSupportPatrols><!--可选项, xs:boolean--></isSupportPatrols>
    <isSupportCombinedPath><!--可选项, xs:boolean, 是否支持云台组合路径-->true</
isSupportCombinedPath>

```

```
</PTZCtrlCap>
<SmartCap/><!--可选项，智能能力-->
<isSupportEhome><!--可选项，xs:boolean--></isSupportEhome>
<isSupportStreamingEncrypt><!--可选项，xs:boolean--></isSupportStreamingEncrypt>
<TestCap>
  <isSupportEmailTest><!--可选项，xs:boolean--></isSupportEmailTest>
</TestCap>
<ThermalCap/><!--可选项，测温能力-->
<WLAlarmCap/><!--可选项，无线报警能力-->
<SecurityCPCapabilities/><!--可选项，报警主机能力-->
<isSupportGIS>
  <!--可选项，xs:boolean，是否支持GIS能力-->
</isSupportGIS>
<isSupportCompass>
  <!--可选项，xs:boolean-->
</isSupportCompass>
<isSupportRoadInfoOverlays>
  <!--可选项，xs:boolean-->
</isSupportRoadInfoOverlays>
<isSupportFaceCaptureStatistics>
  <!--可选项，xs:boolean-->
</isSupportFaceCaptureStatistics>
<isSupportExternalDevice>
  <!--可选项，xs:boolean-->
</isSupportExternalDevice>
<isSupportElectronicsEnlarge>
  <!--可选项，xs:boolean，是否支持电子放大-->
</isSupportElectronicsEnlarge>
<isSupportRemoveStorage>
  <!--可选项，xs:boolean-->
</isSupportRemoveStorage>
<isSupportCloud>
  <!--可选项，xs:boolean-->
</isSupportCloud>
<isSupportRecordHost>
  <!--可选项，xs:boolean-->
</isSupportRecordHost>
<isSupportViewshedOverlay><!--可选项，xs:boolean，是否支持可视域相机目标叠加功能--></
isSupportViewshedOverlay>
<isSupportGPSLabelTracking ><!--可选项，xs:boolean，是否支持GPS标签跟踪--></
isSupportGPSLabelTracking>
<isSupportEagleEye>
  <!--可选项，xs:boolean，是否支持鹰眼系列摄像机-->
</isSupportEagleEye>
<isSupportPanorama>
  <!--可选项，xs:boolean，是否支持全景-->
</isSupportPanorama>
<isSupportFirmwareVersionInfo>
  <!--可选项，xs:boolean，是否支持显示固件版本信息-->
</isSupportFirmwareVersionInfo>
```



```
<isSupportExternalWirelessServer>
  <!--可选项, xs: boolean-->
</isSupportExternalWirelessServer>
<isSupportSetupCalibration>
  <!--可选项, xs:boolean, 是否支持设置校验-->
</isSupportSetupCalibration>
<isSupportGetmutexFuncErrMsg>
  <!--可选项, xs:boolean, 是否支持获取互斥量信息-->
</isSupportGetmutexFuncErrMsg>
<isSupportTokenAuthenticate><!--可选项, xs:boolean--></
isSupportTokenAuthenticate>
<isSupportStreamDualVCA><!--可选项, xs:boolean--></isSupportStreamDualVCA>
<isSupportlaserSpotManual>
  <!--可选项, boolean, 是否支持激光照射配置-->
</isSupportlaserSpotManual>
<isSupportRTMP><!--可选项, xs:boolean--></isSupportRTMP>
<isSupportTraffic><!--可选项, xs:boolean--></isSupportTraffic>
<isSupportLaserSpotAdjustment>
  <!--可选项, boolean, 是否支持调节激光照射尺寸-->
</isSupportLaserSpotAdjustment>
<VideoIntercomCap/><!--可选项, 可视对讲能力-->
<isSupportSafetyCabin>
  <!--可选项, xs:boolean-->
</isSupportSafetyCabin>
<isSupportPEA>
  <!--可选项, xs:boolean, 是否支持一键式报警主机能力-->
</isSupportPEA>
<isSupportCurrentLock>
  <!--可选项, xs:boolean, 是否支持锁定电流配置功能-->
</isSupportCurrentLock>
<isSupportGuardAgainstTheft>
  <!--可选项, xs:boolean, 是否支持设备防盗配置-->
</isSupportGuardAgainstTheft>
<isSupportPicInfoOverlap>
  <!--可选项, xs:boolean, 是否支持图片信息叠加-->
</isSupportPicInfoOverlap>
<isSupportPlay>
  <!--可选项, xs: boolean, 是否支持预览-->
</isSupportPlay>
<isSupportPlayback>
  <!--可选项, xs: boolean, 是否支持回放-->
</isSupportPlayback>
<UHFRFIDReader>
  <!--可选项, UHF RFID 读卡器能力-->
  <isSupportBasicInformation>
    <!--可选项, xs:boolean, 是否支持读卡器基本参数配置-->
  </isSupportBasicInformation>
  <isSupportHardDiskStorageTest>
    <!--可选项, xs:boolean, 是否支持读卡器盘存测试-->
  </isSupportHardDiskStorageTest>
```

```
</UHFRFIDReader>
<isSupportIntelligentStructureAnalysis>
  <!--可选项, xs:boolean, 是否支持结构化智能分析-->
</isSupportIntelligentStructureAnalysis>
<isSupportIntelligentAnalysisEngines>
  <!--可选项, xs:boolean, 是否支持智能检索引擎配置-->
</isSupportIntelligentAnalysisEngines>
<PreviewDisplayNum>
  <!--可选项, xs:integer, 预览窗口个数, 即设备端控制的同时预览窗口的数量。超脑 NVS 由于性能
限制, 当前仅支持单台网络摄像机预览, 且不支持回放-->
</PreviewDisplayNum>
<isSupportBoard opt="true,false">
  <!--可选项, xs:boolean, 是否支持子板相关协议-->
</isSupportBoard>
<ResourceSwitch>
  <workMode opt="4KPreview,educationRecord">
    <!--必选项, xs:string, 设备工作模式: "4KPreview"-4K 预览模式、"educationRecord"-
文教录像模式-->
  </workMode>
</ResourceSwitch>
<isSupportCustomStream><!--可选项, xs:boolean, 是否支持自定义码流--></
isSupportCustomStream>
<isSupportTriggerCapCheck>
  <!--可选项, xs:boolean, 是否支持验证报警联动能力-->
</isSupportTriggerCapCheck>
<isSupportActiveMulticast>
  <!--可选项, xs:boolean, 是否支持多播配置-->
</isSupportActiveMulticast>
<isSupportChannelEventCap>
  <!--可选项, xs:boolean, 是否支持根据通道获取事件能力-->
</isSupportChannelEventCap>
<isSupportPictureServer>
  <!--可选项, xs:boolean, 是否支持图片存储服务-->
</isSupportPictureServer>
<isSupportVideoCompositeAlarm>
  <!--可选项, xs:boolean, 是否支持视频复核报警-->
</isSupportVideoCompositeAlarm>
<isSupportSensorCalibrating>
  <!--可选项, xs:boolean, 是否支持双传感器校验-->
</isSupportSensorCalibrating>
<isSupportChannelEventListCap>
  <!--可选项, xs:boolean, 是否支持获取所有通道的事件能力-->
</isSupportChannelEventListCap>
<VCAResourceChannelsCap>
  <!--可选项, 是否支持通过通道切换至智能检索资源-->
  <ChannelsList>
    <channelsID>
      <!--必选项, xs:integer, 设备支持的通道号-->
    </channelsID>
  </ChannelsList>
```

```
</VCAResourceChannelsCap>
<SensorCap/><!--可选项，智能机柜能力-->
<isSupportSecurityCP/>
  <!--可选项，xs:boolean，是否支持报警主机-->
</isSupportSecurityCP>
<isSupportClientProxyWEB>
  <!--可选项，xs:boolean，是否支持客户端代理远程网络配置："true"，不支持不返回该节点-->
</isSupportClientProxyWEB>
<WEBLocation>
  <!--可选项，string type，网页地址："local"-本地设备、"remote"-远程地址。若没有返回该节点，则网页地址默认为本地设备-->
</WEBLocation>
<deviceId>
  <!--可选项，string，设备ID。如果节点<WEBLocation>为“remote”，则需要返回设备唯一ID用来标识设备唯一对应的web包。该字段由web配置包统一管理，一款设备的多种型号返回相同的设备ID-->
</deviceId>
<isSupportTime>
  <!--可选项，xs:boolean，是否支持时间配置，该字段对于交通产品为必选项，且值为true；其他产品使用isSupportTimeCap字段-->
</isSupportTime>
<isSupportTimeZone>
  <!--可选项，xs:boolean，是否支持夏令时配置-->
</isSupportTimeZone>
<isSupportCityManagement>
  <!--可选项，boolean，只读，是否支持智慧城管-->
</isSupportCityManagement>
<isSupportMixedTargetDetection>
  <!--可选项，xs:boolean，是否支持混合目标检测事件-->
</isSupportMixedTargetDetection>
<isSupportFaceContrastMode>
  <!--可选项，xs:boolean，是否支持人脸图片比对模式-->
</isSupportFaceContrastMode>
<isSupportPictureCaptureComparision>
  <!--可选项，xs:boolean，是否支持摄像机抓拍的人脸图和导入人脸图之间的N:1人脸比对-->
</isSupportPictureCaptureComparision>
<isSupportGPSCalibratation>
  <!--可选项，xs:boolean，是否支持GPS标定能力-->
</isSupportGPSCalibratation>
<isSupportChannelFullEventListCap>
  <!--可选项，xs:boolean，是否支持获取所有通道的事件列表能力-->
</isSupportChannelFullEventListCap>
<isSupportAUXInfoCap>
  <!--可选项，xs:boolean，是否支持获取所有通道的属性能力-->
</isSupportAUXInfoCap>
<isSupportCalibrationFile>
  <!--可选项，xs:boolean，是否支持导入校验文件-->
</isSupportCalibrationFile>
<isSupportDisplayTrajectory>
  <!--可选项，xs:boolean，是否支持显示轨迹-->
</isSupportDisplayTrajectory>
```

```

<maximumSuperPositionTime opt="5,10,20,30">
  <!--依赖项, xs:integer, 显示轨迹的最长时间, 单位:秒, 只有支持显示轨迹时, 该节点有效-->
</maximumSuperPositionTime>
<isSupportUnitConfig>
  <!--可选项, xs:boolean, 是否支持单位配置-->
</isSupportUnitConfig>
<isSupportAutoMaintenance>
  <!--可选项, xs:boolean, 是否支持自动运维管理。若返回该节点且值为“true”则表明支持-->
</isSupportAutoMaintenance>
<isSupportGetLinkSocketIP>
  <!--可选项, xs: boolean, "true,false", 是否支持获取当前连接的 SocketIP-->
</isSupportGetLinkSocketIP>
<isSupportIntelligentSearch>
  <!--可选项, xs:boolean, 是否支持智能检索-->
</isSupportIntelligentSearch>
<IOTCap><!--可选项, xs:boolean, 物联网设备访问能力-->
  <supportChannelNum>
    <!--必选项, xs:integer, 物联网设备支持的通道数-->
  </supportChannelNum>
  <startChannelNo>
    <!--可选项, xs:integer, 初始通道 ID, 若无该节点, 说明初始通道 ID 为 1-->
  </startChannelNo>
  <isSupportlinkageChannelsSearch>
    <!--可选项, boolean, 是否支持联动通道搜索-->
  </isSupportlinkageChannelsSearch>
</IOTCap>
<isSupportEncryption>
  <!--可选项, xs: boolean, 码流加密能力-->
</isSupportEncryption>
<AIDEventSupport opt="abandonedObject, pedestrian, congestion, roadBlock,
construction, trafficAccident, fogDetection, wrongDirection, illegalParking,
SSharpDriving, lowSpeed, dragRacing">
  <!--可选项, xs:string, 支持的交通事件类型: "abandonedObject"-抛洒物、"pedestrian"-
行人检测、"congestion"-拥堵、"roadBlock"-路障、"construction"-施工、"trafficAccident"-
交通事故检测、"fogDetection"-浓雾检测、"wrongDirection"-逆行、"illegalParking"-违停、
"SSharpDriving"-蛇形驾驶检测、"lowSpeed"-低速行驶、"dragRacing"-飙车检测-->
  </AIDEventSupport>
  <TFSEventSupport
opt="illegalParking ,wrongDirection,crossLane, laneChange, vehicleExist, turnRound,
parallelParking, notKeepDistance, notSlowZebraCrossing, overtakeRightSide, lowSpeed,
dragRacing, changeLaneContinuously, SSharpDriving, largeVehicleOccupyLine, jamCrossL
ine">
    <!--可选项, xs:string, 支持的执法事件类型: "illegalParking"-违停、"wrongDirection"-
逆行、"crossLane"-压线、"laneChange"-变道、"vehicleExist"-机占非、"turnRound"-掉头、
"parallelParking"-出入车告警、"notKeepDistance"-为保持车距检测、
"notSlowZebraCrossing"-斑马线未减速检测、"overtakeRightSide"-右侧超车检测、"lowSpeed"-
低速行驶检测、"dragRacing"-飙车检测、"changeLaneContinuously"-连续变道检测、
"SSharpDriving"-蛇形驾驶检测、"largeVehicleOccupyLine"-大车占道检测、 "jamCrossL
ine"-排队跳车-->
  </TFSEventSupport>

```

```
<isVehicleStatisticsSupport>
  <!--可选项, xs: boolean, 是否支持设置交通数据采集参数-->
</isVehicleStatisticsSupport>
<isSupportIntersectionAnalysis>
  <!--可选项, xs: boolean, 是否支持路口分析-->
</isSupportIntersectionAnalysis>
<supportRemoteCtrl
opt="up,down,left,right,enter,menu,num,power,esc,edit,F1,.prev,rec,play,stop,not
Support"/><!--是否支持远程控制-->
<isSptDiagnosis>
  <!--可选项, xs:boolean, 是否支持设备诊断-->
</isSptDiagnosis>
<isSptSerialLogCfg>
  <!--可选项, xs:boolean, 是否支持配置串口日志重定向-->
</isSptSerialLogCfg>
<isSptFileExport>
  <!--可选项, xs:boolean, 是否支持从设备端导出文件-->
</isSptFileExport>
<isSptCertificationStandard>
  <!--可选项, xs:boolean, 是否支持为报警主机配置认证标准-->
</isSptCertificationStandard>
<isSptKeypadLock>
  <!--可选项, xs:boolean, 是否支持锁定键盘-->
</isSptKeypadLock>
<MixedTargetDetection><!--可选项, 设备是否支持在混合目标中识别出特定目标-->
  <isSupportFaceRecognition><!--可选项, xs:boolean, 是否支持人脸识别--></
isSupportFaceRecognition>
  <isSupportHumanRecognition><!--可选项, xs:boolean, 是否支持人体识别--></
isSupportHumanRecognition>
  <isSupportVehicleRecognition><!--可选项, xs:boolean, 是否支持车辆识别--></
isSupportVehicleRecognition>
</MixedTargetDetection>
<isSupportDiscoveryMode><!--可选项, xs:boolean--></isSupportDiscoveryMode>
<streamEncryptionType>
  <!--依赖项, xs:string, 码流加密类型:"RTP/TLS"、"SRTP/UDP"、"SRTP/MULTICAST"。只有
当<isSupportEncryption>值为“true”时该节点有效, 设备端可以支持一种或多种码流加密类型-->
</streamEncryptionType>
<isSupportLms><!--可选项, xs:boolean, 是否支持激光--></isSupportLms>
<isSupportLCDScreen><!--可选项, xs:boolean, 是否支持液晶显示器--></
isSupportLCDScreen>
<isSupportBluetooth><!--可选项, xs:boolean, 是否支持蓝牙, 相关 URI : /ISAPI/System/
Bluetooth/capabilities--></isSupportBluetooth>
<isSupportAcsUpdate>
  <!--可选项, 是否支持升级从门禁设备或外设模块:"true"-支持, 如果没有返回该节点, 则说明不支
持-->
</isSupportAcsUpdate>
<isSupportAccessControlCap>
  <!--可选项, 是否支持门禁能力:"true"-支持, 若没有返回该节点, 则说明不支持-->
</isSupportAccessControlCap>
```

```
<isSupportIDCardInfoEvent><!--可选项，是否支持身份证刷卡事件："true"-支持，若该功能不支持，则不返回该节点--></isSupportIDCardInfoEvent>
<OpenPlatformCap><!--可选项，嵌入开放平台功能，可参见报文 XML_OpenPlatformCap-->
<isSupportInstallationAngleCalibration>
  <!--可选项，xs:boolean，是否支持安装角度校准-->
</isSupportInstallationAngleCalibration>
<isSupportZeroBiasCalibration>
  <!--可选项，xs:boolean，是否支持零偏差校准-->
</isSupportZeroBiasCalibration>
<isSupportDevStatus><!--可选项，xs:boolean，是否支持获取设备状态--></
isSupportDevStatus>
<isSupportRadar><!--可选项，xs:boolean，是否支持安防雷达--></isSupportRadar>
<SHMCap><!--可选项-->
  <isSupportHighHDTemperature><!--可选项，xs:boolean，是否支持硬盘高温检测--></
isSupportHighHDTemperature>
  <isSupportLowHDTemperature><!--可选项，xs:boolean，是否支持硬盘低温检测--></
isSupportLowHDTemperature>
  <isSupportHDImpact><!--可选项，xs:boolean，是否支持硬盘冲击检测--></
isSupportHDImpact>
  <isSupportHDBadBlock><!--可选项，xs:boolean，是否支持硬盘扇区检测--></
isSupportHDBadBlock>
  <isSupportSevereHDFailure><!--可选项，xs:boolean，是否支持硬盘严重故障检测--></
isSupportSevereHDFailure>
</SHMCap>
<isSupportBVCorrect><!--可选项，xs:boolean，是否支持配置摄像机校正参数--></
isSupportBVCorrect>
<guideEventSupport opt="linkageCapture">
  <!--可选项，xs:string，支持按指示进行快速联动的事件："linkageCapture"-抓拍联动-->
</guideEventSupport>
<isSupportAutoSwitch><!--可选项，xs:boolean，是否支持轮巡--> true</
isSupportAutoSwitch>
<isSupportDataPrealarm><!--可选项，xs:boolean，是否支持交通预警事件--></
isSupportDataPrealarm>
<supportGISEvent opt="AID,TPS,ANPR,mixedTargetDetection">
  <!--可选项，xs:string，支持GIS信息接入的事件类型：AID（对应SDK事件：
COMM_ALARM_AID_V41）、TPS（对应SDK事件：COMM_ALARM_TPS_REAL_TIME）、ANPR（对应SDK事件：
COMM_ITS_PLATE_RESULT）、mixedTargetDetection-混合目标检测-->
</supportGISEvent>
<isSupportIntelligentMode><!--可选项，xs:boolean，是否支持智能场景切换（相关URI：/
ISAPI/System/IntelligentSceneSwitch?format=json）--></isSupportIntelligentMode>
<isSupportCertificateCaptureEvent><!--可选项，xs:boolean，是否支持证件抓拍和比对事件：
"true"-是，如果不支持该功能，不返回该节点--></isSupportCertificateCaptureEvent>
<isSupportAlgorithmsInfo><!--可选项，xs:boolean，是否支持获取算法库版本信息：true-是，
如果不支持该功能，不返回该节点--></isSupportAlgorithmsInfo>
<isSupportChannelOccupy><!--可选项，xs:boolean，是否支持室外消防通道被车辆占用检测--></
isSupportChannelOccupy>
<isSupportOffDuty><!--可选项，xs:boolean，是否支持消控室人员离岗检测--></
isSupportOffDuty>
<isSupportNoCertificate><!--可选项，xs:boolean，是否支持消控室持证人员不足检测--></
```

```
isSupportNoCertificate>
  <isSupportSmokeAlarm><!--可选项, xs:boolean, 是否支持烟雾报警--></
isSupportSmokeAlarm>
  <isSupportBatteryCarDisobey><!--可选项, xs:boolean, 是否支持电瓶车违规停放检测--></
isSupportBatteryCarDisobey>
  <isSupportNoFireExtinguisherRecog><!--可选项, xs:boolean, 是否支持灭火器缺失检测--></
isSupportNoFireExtinguisherRecog>
  <isSupportIndoorPasswayBlock><!--可选项, xs:boolean, 是否支持室内通道堵塞检测--></
isSupportIndoorPasswayBlock>
  <isSupportFireSmartFireDetect><!--可选项, xs:boolean, 是否支持火点检测--></
isSupportFireSmartFireDetect>
  <isSupportDetectorRunningStatus><!--可选项, xs:boolean, 是否支持探测器运行状态--></
isSupportDetectorRunningStatus>
  <isSupportDetectorOperationStatus><!--可选项, xs:boolean, 是否支持探测器操作状态--></
isSupportDetectorOperationStatus>
  <isSupportDetectorTemperatureAlarm
opt="highTemperature, riseTemperature, flame"><!--可选项, xs:boolean, 是否支持探测器温
度报警：“highTemperature”-高温报警、“riseTemperature”-温升报警、“flame”-火焰报警--></
isSupportDetectorTemperatureAlarm>
  <isSupportDetectorShelterAlarm><!--可选项, xs:boolean, 是否支持探测器物理遮挡报警--
--></isSupportDetectorShelterAlarm>
  <isSupportDetectorMotionAlarm><!--可选项, xs:boolean, 是否支持探测器物理移动报警--></
isSupportDetectorMotionAlarm>
  <isSupportDetectorTamperAlarm><!--可选项, xs:boolean, 是否支持探测器防拆报警--></
isSupportDetectorTamperAlarm>
  <isSupportDetectorEmergencyAlarm><!--可选项, xs:boolean, 是否支持探测器紧急报警--></
isSupportDetectorEmergencyAlarm>
  <isSupportSmokingDetectAlarm><!--可选项, xs:boolean, 是否支持抽烟报警--></
isSupportSmokingDetectAlarm>
  <isSupportDetectorSmokeAlarm><!--可选项, xs:boolean, 是否支持探测器烟雾报警--></
isSupportDetectorSmokeAlarm>
  <isSupportDetectorCombustibleGasAlarm><!--可选项, xs:boolean, 是否支持探测器可燃气体
报警--></isSupportDetectorCombustibleGasAlarm>
  <isSupportFireControlData><!--可选项, xs:boolean, 是否支持消防实时数据上报--></
isSupportFireControlData>
  <isSupportFireNoRegulation><!--可选项, xs:boolean, 是否支持人离火报警--></
isSupportFireNoRegulation>
  <isSupportSmokeFireRecognize><!--可选项, xs:boolean, 是否支持烟火检测事件上报--></
isSupportSmokeFireRecognize>
  <isSupportPersonArmingTrack><!--可选项, xs:boolean, 是否支持人员布控(相关 URI:/ISAPI/
Intelligent/channels/<ID>/personArmingTrack/capabilities?format=json) --></
isSupportPersonArmingTrack>
  <isSupportManualPersonArmingTrack><!--可选项, xs:boolean, 是否支持手动人员布控(相关
URI:/ISAPI/Intelligent/channels/<ID>/manualPersonArmingTrack?format=json) --></
isSupportManualPersonArmingTrack>
  <isSupportGPSCalibrationMode><!--可选项, xs:boolean, 是否支持 GPS 标定方式(相关 URI:/
ISAPI/System/GPSCalibration/channels/<ID>/mode?format=json) --></
isSupportGPSCalibrationMode>
  <isSupportGPSVerification><!--可选项, xs:boolean, 是否支持 GPS 校验点(相关 URI:/ISAPI/
```

```
System/GPSVerification/channels/<ID>/points?format=json) --></
isSupportGPSVerification>
  <isSupportHBDLib><!--可选项, xs:boolean, 是否支持人体库配置 (相关 URI : /ISAPI/
Intelligent/HBDLib/capabilities?format=json) --></isSupportHBDLib>
  <isSupportDisinfectionInfo><!--可选项, xs:boolean, 是否支持消毒信息上报, 对应事件上报
报文中 eventType:disinfectionInfo--></isSupportDisinfectionInfo>
  <isSupportTempQueryDelete><!--可选项, xs:boolean, 是否支持测温查询删除, 对应事件上报报
文中 eventType:tempQueryDelete--></isSupportTempQueryDelete>
  <isSupportFileSearch><!--可选项, xs:boolean, 是否支持文件查询, 对应 URI : /ISAPI/System/
fileSearch?format=json--></isSupportFileSearch>
  <isSupportFileExportAsync><!--可选项, xs:boolean, 是否支持文件异步导出, 对应 URI : /
ISAPI/System/fileExport/async?format=json--></isSupportFileExportAsync>
  <isSupportSSDFileSystemCapacity><!--可选项, xs:boolean, 是否支持获取 SSD 系统容量分布
(/ISAPI/System/SSDFileSystem/capacity?format=json) --></
isSupportSSDFileSystemCapacity>
  <isSupportFireEscapeDetection><!--可选项, xs:boolean, 是否支持消防通道检测, 对应 URI : /
ISAPI/Intelligent/channels/<ID>/fireEscapeDetection/capabilities?format=json--
></isSupportFireEscapeDetection>
  <isSupportTakingElevatorDetection><!--可选项, xs:boolean, 是否支持乘梯检测, 对应 URI : /
ISAPI/Intelligent/channels/<ID>/takingElevatorDetection/capabilities?
format=json--></isSupportTakingElevatorDetection>
  <isSupportVehicleMatchResult><!--可选项, xs:boolean, 设备是否支持车牌比对, 对应 URI : /
ISAPI/Intelligent/channels/<ID>/licensePlateContrast/capabilities?format=json --
></isSupportVehicleMatchResult>
  <isSupportDataAware><!--可选项, xs:boolean, 数据感知支持能力, 功能对应的接口能力 : /ISAPI/
System/dataAware/capabilities?format=json--> </isSupportDataAware>
  <isSupportAbnormalReboot><!--可选项, xs:boolean, 是否支持异常重启事件--></
isSupportAbnormalReboot>
  <isSupportVisitorEvent><!--可选项, xs:boolean, 是否支持访客事件, 对应事件上报报文 :
JSON_EventNotificationAlert_VisitorEventMsg (eventType 为 “VisitorEvent”) --></
isSupportVisitorEvent>
  <isSupportEmployeeAuthEvent><!--可选项, xs:boolean, 是否支持员工授权事件, 对应事件上报
报文 : JSON_EventNotificationAlert_EmployeeAuthorizationEventMsg (eventType 为
“EmployeeAuthEvent”) --></isSupportEmployeeAuthEvent>
  <isSupportVisitorInfoSearchEvent><!--可选项, xs:boolean, 是否支持访客信息查询事件, 对
应事件上报报文 : JSON_EventNotificationAlert_VisitorInfoSearchEventMsg (eventType 为
“VisitorInfoSearchEvent”) --></isSupportVisitorInfoSearchEvent>
  <isSupportVehicleMatchResult><!--可选项, xs:boolean, 设备是否支持车牌比对 (相关 URI : /
ISAPI/Intelligent/channels/<ID>/licensePlateContrast/capabilities?format=json) --
></isSupportVehicleMatchResult>
  <isSupportFireEscapeDetection><!--可选项, xs:boolean, 是否支持消防通道检测 (相关 URI : /
ISAPI/Intelligent/channels/<ID>/fireEscapeDetection/capabilities?format=json) --
></isSupportFireEscapeDetection>
  <isSupportTakingElevatorDetection><!--可选项, xs:boolean, 是否支持乘梯检测 (相关 URI : /
ISAPI/Intelligent/channels/<ID>/takingElevatorDetection/capabilities?
format=json) --></isSupportTakingElevatorDetection>
  <isSupportCartridgeCapacity><!--可选项, xs:boolean, 是否支持墨盒容量提醒配置 (相关
URI : /ISAPI/System/printer/cartridgeCapacity/capabilities?format=json) --></
isSupportCartridgeCapacity>
```



```
<isSupportCDCakeBoxCapacity><!--可选项, xs:boolean, 是否支持光盘容量提醒 (相关 URI : /ISAPI/System/printer/CDCakeBoxCapacity/capabilities?format=json) --></isSupportCDCakeBoxCapacity>
<isSupportSSDFileSystemUpgrade><!--可选项, xs:boolean, 是否支持是否支持 SSD 文件系统升级 (相关 URI : /ISAPI/System/SSDFileSystem/upgrade?format=json) --></isSupportSSDFileSystemUpgrade>
<isSupportSSDFileSystemFormat><!--可选项, xs:boolean, 是否支持是否支持 SSD 格式化 (相关 URI : /ISAPI/System/SSDFileSystem/format?format=json) --></isSupportSSDFileSystemFormat>
<isSupportSSDFileSystemCapacity><!--可选项, xs:boolean, 是否支持获取 SSD 系统容量分布 (相关 URI : /ISAPI/System/SSDFileSystem/capacity?format=json) --></isSupportSSDFileSystemCapacity>
<isSupportAIOpenPlatform> <!--可选项, xs:boolean, 是否支持 AI 开放平台设备级能力, 支持则返回 true, 不支持不返回该节点--></isSupportAIOpenPlatform>
<isSupportPictureDownloadError><!--可选项, xs:boolean, 是否支持图片下载失败异常上报--></isSupportPictureDownloadError>
<isSupportConsumptionEvent><!--可选项, xs:boolean, 是否支持消费事件, 对应事件上报报文的 eventType 为 "ConsumptionEvent" --></isSupportConsumptionEvent>
<isSupportTransactionRecordEvent><!--可选项, xs:boolean, 是否支持交易记录事件, 对应事件上报报文的 eventType 为 "TransactionRecordEvent" --></isSupportTransactionRecordEvent>
<isSupportDepositRetrieveEvent><!--可选项, xs:boolean, 是否支持智能柜物品存取事件, 对应事件上报报文中 eventType 为 "DepositRetrieveEvent" --></isSupportDepositRetrieveEvent>
<isSupportDepositTimeoutEvent><!--可选项, xs:boolean, 是否支持智能柜物品寄存超时事件, 对应事件上报报文中 eventType 为 "DepositTimeoutEvent" --></isSupportDepositTimeoutEvent>
<isSupportCabinetStatusEvent><!--可选项, xs:boolean, 是否支持智能柜状态变化事件, 对应事件上报报文中 eventType 为 "CabinetStatusEvent" --></isSupportCabinetStatusEvent>
<isSupportCabinetAuthorityExpired><!--可选项, xs:boolean, 是否支持智能柜权限过期事件, 对应事件上报报文中 eventType 为 "CabinetStatusEvent" --></isSupportCabinetAuthorityExpired >
<isSupportISUPHttpPassthrough><!--可选项, xs:boolean, 是否支持 ISUPV5.0http 透传, 不返回该节点表示不支持--></isSupportISUPHttpPassthrough>
<isSupportAppCfg><!--可选项, xs:boolean, 是否支持应用管理 (相关 URI : /ISAPI/System/app/capabilities?format=json) --></isSupportAppCfg>
<isSupportCleanup><!--可选项, xs:boolean, 是否支持一键清理 (相关 URI : /ISAPI/System/cleanup?format=json) --></isSupportCleanup>
<isSupportEndUserCheckMsg
opt="shutdown, reboot, factoryReset, cleanup, parentPickUpQuery, schoolAttendanceQuery"><!--可选项, xs:boolean, 是否支持最终用户确认信息上报: "shutdown" (关机)、"reboot" (重启)、"factoryReset" (恢复出厂)、"cleanup" (一键清理)、"parentPickUpQuery" (家长接送信息查询)、"schoolAttendanceQuery" (校园考勤信息查询)。对应事件上报报文中事件 (eventType) : endUserCheckMsg--></isSupportEndUserCheckMsg>
<isSupportScreenConfig><!--可选项, xs:boolean, 是否支持屏幕参数配置 (相关 URI : /ISAPI/System/screenConfig/capabilities?format=json) --></isSupportScreenConfig>
<isSupportTemperatureMonitor><!--可选项, xs:boolean, 是否支持温度监测参数配置 (相关 URI : /ISAPI/System/temperatureMonitor/capabilities?format=json) --></isSupportTemperatureMonitor>
```

```
isSupportTemperatureMonitor>
  <isSupportSignalSource><!--可选项, xs:boolean, 是否支持信号源参数配置 (相关 URI : /ISAPI/
System/signalSource/capabilities?format=json) --></isSupportSignalSource>
  <isSupportUSBConfig><!--可选项, xs:boolean, 是否支持 USB 参数配置 (相关 URI : /ISAPI/
System/USBConfig/capabilities?format=json) --></isSupportUSBConfig>
  <isSupportStandbyStrategy><!--可选项, xs:boolean, 是否支持待机策略参数配置 (相关 URI : /
ISAPI/System/standbyStrategy/capabilities?format=json) --></
isSupportStandbyStrategy>
  <isSupportWakeupStrategy><!--可选项, xs:boolean, 是否支持唤醒策略参数配置 (相关 URI : /
ISAPI/System/wakeupStrategy/capabilities?format=json) --></
isSupportWakeupStrategy>
  <isSupportPowerSupply><!--可选项, xs:boolean, 是否支持对外供电, (相关 URL : /ISAPI/
System/powerSupply/capabilities?format=json) --></isSupportPowerSupply>
  <isSupportSleep><!--可选项, xs:boolean, 是否支持休眠配置, 支持返回 true, 不支持不返回--
--></isSupportSleep>
  <isSupportObjectServer><!--可选项, xs:boolean, 是否支持对象存储服务 (相关 URI : /ISAPI/
System/objectServer/capabilities?format=json) --></isSupportObjectServer>
  <isSupportLensParamFile><!--可选项, xs:boolean, 是否支持镜头参数文件导入导出--></
isSupportLensParamFile>
  <isSupportWOL>
    <!--可选项, xs:boolean, 是否支持通过 Wake On-Lan 实现远程开关机 ; 对应 URI (/ISAPI/
System/WOL/capabilities?format=json) -->true
  </isSupportWOL>
  <isSupportSecurityChecking><!--可选项, xs:boolen, 是否支持安检过包检测配置, 支持返回
true, 不支持无需返回--></isSupportSecurityChecking>
  <isSupportMixedSecurityChecking><!--可选项, xs:boolean, 是否支持安检混合检测事件上传--
--></isSupportMixedSecurityChecking>
  <isSupportConsumptionQuery><!--可选项, xs:boolean, 是否支持消费查询请求上报, 事件类型
(eventType) : ConsumptionQuery--></isSupportConsumptionQuery>
  <isSupportWaterQualitySearch><!--可选项, xs:boolean, 是否支持水质数据查询功能 : true
(支持)、false 或不返回代表不支持。相关 URI : /ISAPI/Event/waterQualityDetection/search?
format=json--></isSupportWaterQualitySearch>
  <isSupportMEF><!--可选项, xs:boolean, 是否支持移动电子围栏数据监测。相关 URI : /ISAPI/
System/Network/MEF/<ID>/capabilities?format=json--></isSupportMEF>
  <isSupportCounting><!--可选项, boolean, 只读, 是否支持人数统计--></isSupportCounting>
  <isSupportFramesPeopleCounting><!--可选项, boolean, 只读, 是否支持区域人数统计事件--
--></isSupportFramesPeopleCounting>
  <isSupportInternalParamsAdjustment><!--可选项, xs:boolean, 是否支持按区域进行内参校
正 : true (支持)、false 或不返回 (不支持), 相关 URI : /ISAPI/System/internalParameters/
regionAdjustment/capabilities?format=json--></isSupportInternalParamsAdjustment>
  <isSupportDiagnosticReport><!--可选项, xs:boolean, 是否支持诊断报告功能 : true (支持)、
false 或不返回 (不支持)。相关 URI : /ISAPI/System/diagnosticReport/capabilities?
format=json--></isSupportDiagnosticReport>
  <isSupportWaterQualitySearch><!--可选项, xs:boolean, 是否支持水质数据查询功能 : true
(支持)、false 或不返回 (不支持)。相关 URI : /ISAPI/Event/waterQualityDetection/search?
format=json--></isSupportWaterQualitySearch>
  <isSupportMethaneConcentrationException>
    <!--可选项, boolean, 只读, 是否支持甲烷浓度异常事件 (methaneConcentrationException)--
-->true
```

```

</isSupportMethaneConcentrationException>
<isSupportMethaneLightIntensityException>
  <!--可选项, boolean, 只读, 是否支持甲烷光强异常事件 (methaneLightIntensityException)-->
  <true>
  </isSupportMethaneLightIntensityException>
  <DeployCap><!--可选项, object, 布防非订阅参数能力-->
    <deployID opt="1,2"><!--可选项, xs:integer, 布防类型: 0 (客户端布防)、1 (实时布防)-->
  </deployID>
  </DeployCap>
  <deviceStatus
opt="calling,noCall,medicationChangePrompt,alarming,offline,free"><!--可选项,
xs:string, 设备状态查询条件: "calling"-通话中、"noCall"-非通话状态、
"medicationChangePrompt"-换药提醒中、"alarming"-报警中、"offline"-离线、"free"-空闲。多个
条件默认采用“与”关系--></deviceStatus>
  <logicRelationship opt="and,or"><!--依赖项, xs:string, 设备状态查询条件逻辑关系:
"and"-与关系、"or"-或关系, 默认为"and", 当 deviceStatus 存在时有效--></logicRelationship>
  <alarmType opt="SOS,toilet"><!--依赖项, xs:string, 报警类型: "SOS"-SOS 呼救、"toilet"-
卫生间报警, 当 deviceStatus 存在"alarming"时可选有效--></alarmType>
  <callWaitingType opt="normal,cisit"><!--依赖项, 报警类型: "normal"-正常呼叫等待、
"visit"-探视呼叫等待, 当 deviceStatus 存在"callWaiting"时可选有效--></callWaitingType>
  <periodNumber min="" max=""/><!--可选项, xs:integer, 期号-->
  <buildingNumber min="" max=""/><!--可选项, xs:integer, 楼号-->
  <unitNumber min="" max=""/><!--可选项, xs:integer, 单元号-->
  <deviceIndex min="1" max="15"/><!--可选项, xs:string, 设备序号字符长度取值范围-->
  <industryType opt="builidings,prison,medicalTreatment,broadcasting"><!--可选项,
string, 行业(场景)类型: "builidings"-楼宇、"prison"-监所、"medicalTreatment"-医疗、
"broadcasting"-广播--></industryType>
  <priority min="1" max="100"/><!--可选项, integer, 呼叫优先级-->
  <transportType opt="unicast,multicast"><!--可选项, string, 广播方式: "unicast"-单
播、"multicast"-组播--></transportType>
  <audioVolume min="0" max="15"><!--可选项, string, 播放音量取值范围: [0,15], 默认为 5-->
  </audioVolume>
  <callForwardingEnabled><!--可选项, string, 呼叫转移使能: true-开启、false-关闭, 默认关
闭--></callForwardingEnabled>
  <callForwardingType opt="unconditional,busy,noReply,noReachable,calling"><!--
可选项, string, 呼叫转移类型: "unconditional"-无条件呼叫转移、"busy"-遇忙呼叫转移、"noReply"-
无应答呼叫转移、"noReachable"-不可达呼叫转移、"calling"-通话转移--></callForwardingType>
  <isSupportCallTerminalNumbe><!--可选项, boolean, 是否支持转移目标管理机长号配置-->
  </isSupportCallTerminalNumbe>
  <isSupportMoreFeatures><!--可选项, boolean, 是否接入设备更多功能设置, 不传此节点默认不支
持--></isSupportMoreFeatures>
  <unitType opt="outdoor,indoor,manage,fence,terminal,netAudio"><!--可选项,
string, 设备类型: "terminal"-终端、"netAudio"-网络音响--></unitType>
  <serialNumber min="1" max="32"><!--可选项, string, 设备序列号字符长度取值范围--></
serialNumber>
  <isSupportCityCode><!--可选项, boolean, 是否支持城市编码获取, 相关 URI: /ISAPI/System/
cityCode?format=json--></isSupportCityCode>
  <customCommunicationName min="0" max="128"><!--可选项, string, 自定义通讯名称, 当有

```

此节点时, 设备呼叫时优先显示自定义通讯名称, 而不是终端长号--><customCommunicationName>
 <isSupportMultiDeviceID><!--可选项, xs:boolean, 是否支持多长号配置, 相关 URI : /ISAPI/VideoIntercom/deviceId/batchCfg/capabilities?format=json--></isSupportMultiDeviceID>
 <isNotSupportMultiDeviceIDCustomName><!--可选项, xs:boolean, 是否不支持多个设备编号自定义名称配置 : true-不支持、false 或者不填-支持, 不支持时, 用户无法通过多长号配置接口给设备配置自定义名称--></isNotSupportMultiDeviceIDCustomName>
 <isSupportBroadcastTriggerEvent><!--可选项, xs:boolean, 只读, 是否支持广播触发事件, 寻呼话筒上报该事件到平台后, 平台根据上报内容中的终端号列表进行广播并进行确认, 事件类型 : broadcastTriggerEvent, 确认 URI : /ISAPI/VideoIntercom/broadcastTriggerConfirm?format=json->true</isSupportBroadcastTriggerEvent>
 <isSupportAutoReboot><!--可选项, xs:boolean, 是否支持自动重启--></isSupportAutoReboot>
 <isSupportFactoryReset><!--可选项, xs:boolean, 是否支持恢复出厂设置。交通产品, 该节点为必选项--></isSupportFactoryReset>
 <isSupportGaugeReading>
 <!--可选项, xs:boolean, 是否支持水尺识别功能, 相关 URI : GET /ISAPI/Event/channels/<channelID>/gaugeReading/capabilities?format=json。不支持不返回该节点-->true
 </isSupportGaugeReading>
 <isSupportUncertificateCompareEvent>
 <!--可选项, xs:boolean, 是否支持无证比对事件, 对应事件类型 (**eventType**) 为“UncertificateCompareEvent”, 不支持不返回-->true
 </isSupportUncertificateCompareEvent>
 </DeviceCap>

16.4.15 XML_EventNotificationAlert_AlarmInputAlarmMsg

关于报警输入报警信息的 XML 报文

```
<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型:HTTP, HTTPS, EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道编号--></channelID>
  <dateTime>
    <!--必选项, xs:datetime, 事件发生时间, ISO 8601 时间格式, 如:
    2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型, 此处应设为"IO"--></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件):“active”、“inactive” --></eventState>
```

```

<eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
<inputIOPortID><!--可选项, xs:integer, 本地报警输入 ID--></inputIOPortID>
<dynInputIOPortID><!--可选项, xs:integer, 动态报警输入 ID--></dynInputIOPortID>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
<Extensions>
  <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
  <eventPush><!--必选项, xs:string, 事件或报警推送消息--></eventPush>
</Extensions>
</EventNotificationAlert>

```

16.4.16 XML_EventNotificationAlert_ANPRMsg

关于自动车牌识别事件详情的 XML 报文

```

<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IP 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 端口号--></portNo>
  <protocol><!--可选项, xs:string, 协议类型: HTTP, HTTPS, EHome--></protocol>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
  一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
  relatedChannelList>
  <dateTime><!--必选项, xs:datetime, ISO 8601 时间格式, 例如,
  2017-04-22T15:39:01+08:00--></dateTime>
  <activePostCount><!--必选项, xs:integer--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型: 此处为 "ANPR" --></eventType>
  <eventState><!--必选项, xs:string, 事件状态: active-已发生, inactive-未发生--></
  eventState>
  <eventDescription><!--必选项, xs:string--></eventDescription>
  <channelName><!--可选项, xs:string, 通道名称--></channelName>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <ANPR><!--依赖项, 该节点只有当<eventType>为 "ANPR" -->
    <region><!--可选项, xs:string, 地区, 详见区域码--></region>
    <country><!--可选项, xs:string, 国家和地区, 详见国家和地区代码--></country>
    <area><!--可选项, xs:string, 区域 (省份), 各国家内部区域枚举--></area>
    <licensePlate><!--必选项, xs:string, 车牌号码, 需要带车牌颜色, 例如: "蓝浙 AX2P12" --
    ></licensePlate>
    <line><!--必选项, xs:integer, 已识别的车道号--></line>
    <direction><!--可选项, xs:string, 车牌识别方向: "reverse" -背向, "forward" -前向,
    "unknown" -未知--></direction>
    <confidenceLevel><!--必选项, xs:integer, 置信度等级, 其取值范围从 0 到 100--></
    confidenceLevel>

```

```
<plateType><!--可选项, xs:string, 车牌类型:“unknown”-未知, “92TypeCivil”-92 式民
用车, “arm”-警车, “upDownMilitay”-军用车(上下式), “92TypeArm”-92 式警车,
“leftRightMilitay”-军用车(左右式), “02TypePersonalized”-02 式自定义车辆,
“yellowTwoLine”-黄色双行车的后车牌, “04NewMilitay”-04 式新式军用车, “embassy”-使馆车,
“oneLineArm”-新式武警车(单行), “twoLineArm”-新式武警车(双行), “yellow1225FarmVehicle”-
黄色 1225 结构农用车, “green1325FarmVehicle”-绿色 1325 结构农用车,
“yellow1325FarmVehicle”-黄色 1325 结构农用车, “motorola”-摩托车, “coach”-教练车,
“tempTravl”-临时车, “trailer”-拖车, “consulate”-领馆车, “hongKongMacao”-香港澳门出入
车辆, “tempEntry”-临时进入车辆, “civilAviation”-民航车牌, “newEnergy”-新能源车牌--></
plateType>
<plateColor>
  <!--可选项, xs:string, 车牌颜色:“white”-白色, “yellow”-黄色, “blue”-蓝色, “black”-
黑色, “green”-绿色, “civilAviationBlack”-民航黑色, “civilAviationGreen”-民航绿色,
“newEnergyGreen”-新能源绿色, “newEnergyYellowGreen”-新能源黄绿色, “other”-其他颜色-->
</plateColor>
<licenseBright>
  <!--可选项, xs:integer, 车牌图片亮度, 其取值范围从 0 到 255-->
</licenseBright>
<Rect>
  <!--可选项, 匹配图片中车牌坐标-->
  <height>
    <!--必选项, xs:float, 高度-->
  </height>
  <width>
    <!--必选项, xs:float, 宽度-->
  </width>
  <x>
    <!--必选项, xs:float, 左上角起始点的 X 坐标-->
  </x>
  <y>
    <!--必选项, xs:float, 左上角起始点的 Y 坐标-->
  </y>
</Rect>
<pilotsafebelt>
  <!--可选项, xs:string, 驾驶员安全带是否已系:“unknown”-未知, “yes”-是, “no”-否-->
</pilotsafebelt>
<vicepilotsafebelt>
  <!--可选项, xs:string, 副驾驶员安全带是否已系:“unknown”-未知, “yes”-是, “no”-否-->
</vicepilotsafebelt>
<pilotsunvisor>
  <!--可选项, xs:string, 驾驶室遮阳板是否打开:“unknown”-未知, “yes”-是, “no”-否-->
</pilotsunvisor>
<vicepilotsunvisor>
  <!--必选项, xs:string, 副驾驶室遮阳板是否打开:“unknown”-未知, “yes”-是, “no”-否-->
</vicepilotsunvisor>
<envprosign>
  <!--可选项, xs:string, 是否为黄标车检:“unknown”-未知, “yes”-是, “no”-否-->
</envprosign>
<dangmark>
```

```
<!--可选项, xs:string, 是否为危险品车:“unknown”-未知, “yes”-是, “no”-否-->
</dangmark>
<uphone>
  <!--可选项, xs:string, 驾驶员是否在打电话:“unknown”-未知, “yes”-是, “no”-否-->
</uphone>
<pendant>
  <!--可选项, xs:string, 车窗是否悬挂物:“unknown”-未知, “yes”-是, “no”-否-->
</pendant>
<tissueBox>
  <!--可选项, xs:string, 是否放置纸巾盒:“unknown”-未知, “yes”-是, “no”-否-->
</tissueBox>
<frontChild>
  <!--可选项, xs:string, 副驾驶人员是否怀抱婴儿:“unknown”-未知, “yes”-是, “no”-否-->
</frontChild>
<label>
  <!--可选项, xs:string, 是否贴标签:“unknown”-未知, “yes”-是, “no”-否-->
</label>
<decoration>
  <!--可选项, xs:string, 车身是否装饰:“unknown”-未知, “yes”-是, “no”-否-->
</decoration>
<smoking>
  <!--可选项, xs:string, 是否抽烟:“unknown”-未知, “yes”-是, “no”-否-->
</smoking>
<perfumeBox>
  <!--可选项, xs:string, 是否放置香水瓶:“unknown”-未知, “yes”-是, “no”-否-->
</perfumeBox>
<pdvs>
  <!--可选项, xs:string, 汽车天窗是否有探出身体:“unknown”-未知, “yes”-是, “no”-否-->
</pdvs>
<helmet>
  <!--可选项, xs:string, 是否戴头盔:“unknown”-未知, “yes”-是, “no”-否-->
</helmet>
<twoWheelVehicle>
  <!--可选项, xs:string, 是否为两轮车:“unknown”-未知, “yes”-是, “no”-否-->
</twoWheelVehicle>
<threeWheelVehicle>
  <!--可选项, xs:string, 是否为三轮车:“unknown”-未知, “yes”-是, “no”-否-->
</threeWheelVehicle>
<blackness>
  <!--可选项, xs:integer, 林格曼黑度值, 用于冒黑烟检测-->
</blackness>
<plateCharBelieve>
  <!--可选项, xs:string, 识别车牌各个字符的置信度。如检测到车牌"A12345", 置信度为
10, 20, 30, 40, 50, 60, 70, 则表示"浙"字正确的可能性只有 10%, "A"字的正确的可能性是 20%-->
</plateCharBelieve>
<speedLimit>
  <!--可选项, xs:integer, 最大速度限制, 该节点只有在超速时有效-->
</speedLimit>
<illegalInfo>
  <!--可选项, 违章信息-->
```

```

<illegalCode>
  <!--必选项, xs:string, 违章代码-->
</illegalCode>
<illegalName>
  <!--必选项, xs:string, 违章名称-->
</illegalName>
<illegalDescription>
  <!--可选项, xs:string, 违章描述-->
</illegalDescription>
</illegalInfo>
<vehicleType>
  <!--可选项, xs:string, 车辆类型：“unknown”-未知, “largeBus”-公交, “truck”-卡车,
“vehicle”-小汽车, “van”-, 面包车, “buggy”-四驱车, “pedestrian”-行人,
“twoWheelVehicle”-两轮车, “threeWheelVehicle”-三轮车, “SUVMPV”-越野车, “mediumBus”-
客车, “motorVehicle”-摩托车, “nonmotorVehicle”-非机动车, “smallCar”-小轿车, “miniCar”-
迷你车, “pickupTruck”-皮卡车-->
</vehicleType>
<postPicFileName>
  <!--可选项, xs:string, 当违章发生时, 选择图片作为卡口图, “none”-不选择-->
</postPicFileName>
<featurePicFileName>
  <!--可选项, xs:string, 当电警检测到闯红灯时, 选择图片作为近景图, “none”-不选择-->
</featurePicFileName>
<detectDir>
  <!--可选项, xs:integer, 侦测方向：1-上行, 2-下行, 3-双向, 4-向西, 5-向北, 6-向东, 7-
向南, 8-其他-->
</detectDir>
<detectType>
  <!--可选项, xs:integer, 侦测类型：0-车辆检测, 1-地感触发, 2-视频触发, 3-多帧识别, 4-雷达
触发, 5-混行检测-->
</detectType>
<barrierGateCtrlType>
  <!--可选项, xs:integer, 是否启用历史数据：0-启用, 1-不启用-->
</barrierGateCtrlType>
<alarmDataType>
  <!--可选项, xs:integer, 0-实时数据, 1-历史数据-->
</alarmDataType>
<dwIllegalTime>
  <!--可选项, xs:integer, 违章持续时长, 即为最后图片的抓拍时间和首张图片的抓拍时间的差值,
单位：毫秒-->
</dwIllegalTime>
<vehicleInfo>
  <index>
    <!--必选项, xs:integer, 车辆编号-->
  </index>
  <vehicleType>
    <!--可选项, xs:integer, 触发类型：0-其他车辆, 1-小型车, 2-大型车, 3-行人触发, 4-两
轮车, 5-三轮车, 6-机动车-->
  </vehicleType>
  <colorDepth>

```



```
<!--必选项, xs:integer, 车辆颜色深度:0-深色, 1-浅色-->
</colorDepth>
<color>
  <!--必选项, xs:string, 车辆颜色:“unknown”-未知, “white”-白, “silver”-银,
“gray”-灰, “blacks”-黑, “red”-红, “deepBlue”-深蓝, “blue”-蓝, “yellow”-黄, “green”-
绿, “brown”-棕, “pink”-粉色, “purple”-紫色, “deepGray”-深灰, “cyan”-青色, “orange”-
桔色-->
</color>
<speed>
  <!--必选项, xs:integer, 车辆速度, 单位:km/h-->
</speed>
<length>
  <!--必选项, xs:integer, 前辆车的车身长度, 单位:分米-->
</length>
<vehicleLogoRecog>
  <!--必选项, xs:integer, 车辆母品牌-->
</vehicleLogoRecog>
<vehileSubLogoRecog>
  <!--可选项, xs:integer, 车辆子品牌-->
</vehileSubLogoRecog>
<vehileModel>
  <!--可选项, xs:integer, 车辆子品牌的型号-->
</vehileModel>
<vehicleTypeByWeight>
  <!--可选项, xs:integer, 1-一类车(客车:7座及7座以下、货车:2吨以下), 2-二类车(客
车:8-19座、货车:载重2吨-5吨(含)), 3-三类车(客车:20-39座、货车:载重5吨-10吨(含)),
4-四类车(客车:40座以上、货车:10吨-15吨(含)), 5-五类车(货车:载重15吨以上)-->
</vehicleTypeByWeight>
</vehicleInfo>
<EntranceInfo>
  <!--opt-->
  <parkingID>
    <!--可选项, xs:string, 停车位编号-->
  </parkingID>
  <gateID>
    <!--可选项, xs:string, 出入口编号-->
  </gateID>
  <direction>
    <!--可选项, xs:string, 进出方向-->
  </direction>
  <cardNo>
    <!--可选项, xs:string, 卡号-->
  </cardNo>
  <parkType>
    <!--可选项, xs:string, 停车类型:“permanent”-固定, “temporary”-临时-->
  </parkType>
</EntranceInfo>
<pictureInfoList><!--req-->
  <pictureInfo><!--必选项, xs:list-->
    <fileName>
```

```

    <!--必选项, xs:string, 图片名称, 与报警图片名称对应-->
  </fileName>
  <type>
    <!--必选项, xs:string, 图片类型:“detectionPicture”-侦测图片,
“licensePlatePicture”-车牌图片,“pilotPicture”-驾驶员图片,“copilotPicture”-副驾驶驾驶员图
片,“compositePicture”-合成图片,“plateBinaryPicture”-车牌二进制图片,
“nonMotorPicture”-非机动车图片,“pedestrianDetectionPicture”-行人侦测图片,
“pedestrianPicture”-行人图片-->
  </type>
  <dataType>
    <!--必选项, xs:integer, 数据类型:0-数据,1-URL-->
  </dataType>
  <picRecogMode>
    <!--可选项, xs:integer, 0-前向车牌识别,1-后向车牌识别-->
  </picRecogMode>
  <redLightTime>
    <!--可选项, xs:integer, 经过的红灯时长, 单位:秒-->
  </redLightTime>
  <vehicleHead>
    <!--可选项, xs:integer, "unknown"-未知,“forward”-前向车牌识别,“back”-后向
车牌识别-->
  </vehicleHead>
  <absTime>
    <!--可选项, xs:time, 绝对时间, 格式:yyyyMMddHHmmssxxx, 例如,
20090810235959999, 最后三位数字表示毫秒-->
  </absTime>
  <plateRect>
    <!--依赖项, 当前画面百分比大小乘以1000的归一化值, 精确到小数点后三位, 该节点在
<type>为“detectionPicture”时有效-->
  <X>
    <!--必选项, xs:integer, 边框左上角的X坐标-->
  </X>
  <Y>
    <!--必选项, xs:integer, 边框左上角的Y坐标-->
  </Y>
  <width>
    <!--必选项, xs:integer, 边框的宽度-->
  </width>
  <height>
    <!--必选项, xs:integer, 边框的高度-->
  </height>
  </plateRect>
  <vehicle1Rect>
    <!--依赖项, 当前画面百分比大小乘以1000的归一化值, 精确到小数点后三位, 该节点在
<type>为“detectionPicture”时有效-->
  <X>
    <!--必选项, xs:integer, 边框左上角的X坐标-->
  </X>
  <Y>

```

```

    <!--必选项, xs:integer, 边框左上角的 Y 坐标-->
  </Y>
  <width>
    <!--必选项, xs:integer, 边框的宽度-->
  </width>
  <height>
    <!--必选项, xs:integer, 边框的高度-->
  </height>
</vehicleRect>
<pictureURL>
  <!--依赖项, xs:string, 图片 URL, 该节点只有在<dataType>为“URL”时有效-->
</pictureURL>
  <pId><!--可选项, xs:string, 字符串长度为 32 字节, 建议生成规则 (设备序列号+设备启动
时间至今的时间+随机数) --></pId>
</pictureInfo>
</pictureInfoList>
<hasMoreData>
  <!--可选项, xs:boolean, 是否存在更多数据。该节点用于上报车牌信息和完整的 XML 及图片数
据。完整数据和车牌信息通过 UUID 关联-->
</hasMoreData>
  <originalLicensePlate><!--可选项, xs:string, 原始车牌号码--></
originalLicensePlate>
  <CRIndex><!--可选项, xs:integer, 国家/地区索引, 为 0 则代表通用, 不区分具体国家/地区, 包
含 countryIndex, 扩展了区域, 参考国家和地区代码, 优先使用该字段--></CRIndex>
</ANPR>
<UUID>
  <!--可选项, xs:string, 通用 ID, 用于跨服务器关联同一次抓拍-->
</UUID>
<picNum>
  <!--可选项, xs:integer, 图片数量-->
</picNum>
<monitoringSiteID>
  <!--可选项, xs:string, 相机编号-->
</monitoringSiteID>
<ePlateUUID>
  <!--可选项, xs:string, 电子车牌 ID, 如果这个节点存在且被赋值, 说明有关联的电子车牌--->
</ePlateUUID>
</EventNotificationAlert>

```

参见

[区域码](#)

[国家和地区代码](#)

16.4.17 XML_EventNotificationAlert_audioException

关于音频异常侦测报警信息的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型:HTTP、HTTPS、EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道编号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
  一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
  relatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, 事件发生时间, ISO 8601 时间格式, 如 :
    2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型, 此处应设为"audioexception"--></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): "active"、"inactive" --></
  eventState>
  <eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
  <channelName><!--可选项, xs:string, 通道名称--></channelName>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <Extensions>
    <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
    <eventPush><!--必选项, xs:string, 事件或报警推送消息--></eventPush>
  </Extensions>
</EventNotificationAlert>

```

16.4.18 XML_EventNotificationAlert_DefocusDetectionAlarmMsg

关于虚焦侦测报警信息的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型:HTTP、HTTPS、EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道编号--></channelID>
  <relatedChannelList>
    <!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道), 用于
    平台收到报警时展示预览或回放, 多个通道号用逗号隔开-->
  </relatedChannelList>

```

```

</relatedChannelList>
<dateTime>
  <!--必选项，xs:datetime，事件发生时间，ISO 8601 时间格式，如：
2017-04-22T15:39:01+08:00-->
</dateTime>
<activePostCount><!--必选项，xs:integer，事件发生次数--></activePostCount>
<eventType><!--必选项，xs:string，事件类型，此处应设为"defocus"--></eventType>
<eventState><!--必选项，xs:string，事件状态（持续事件）：“active”、“inactive”--></
eventState>
<eventDescription><!--必选项，xs:string，事件描述--></eventDescription>
<channelName><!--可选项，xs:string，通道名称--></channelName>
<deviceID><!--可选项，xs:string，设备 ID--></deviceID>
<Extensions>
  <serialNumber><!--必选项，xs:string，序列号--></serialNumber>
  <eventPush><!--必选项，xs:string，事件或报警推送消息--></eventPush>
</Extensions>
</EventNotificationAlert>

```

16.4.19 XML_EventNotificationAlert_FaceDetectionAlarmMsg

关于人脸侦测事件详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项，xs:string，IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项，xs:string，IPv6 地址--></ipv6Address>
  <portNo><!--可选项，xs:integer，设备通信端口号--></portNo>
  <protocolType><!--可选项，xs:string，传输协议类型：HTTP，HTTPS--></protocol>
  <macAddress><!--可选项，xs:string，MAC 地址--></macAddress>
  <dynChannelID><!--可选项，xs:string，数字通道 ID--></dynChannelID>
  <channelID><!--依赖项，xs:string，通道 ID--></channelID>
  <relatedChannelList><!--可选项，xs:string，报警关联（同源）通道列表（与 channelID 在同
一个相机上的通道），用于平台收到报警时展示预览或回放，多个通道号用逗号隔开--></
relatedChannelList>
  <dateTime><!--必选项，xs:datetime，ISO 8601 时间格式，例如，
2017-04-22T15:39:01+08:00--></dateTime>
  <activePostCount><!--必选项，xs:integer--></activePostCount>
  <eventType><!--必选项，xs:string，事件类型：此处需要设为“facedetection”--></
eventType>
  <eventState><!--必选项，xs:string，事件状态（持续事件）：active-已发生，inactive-未发
生--></eventState>
  <eventDescription><!--必选项，xs:string--></eventDescription>
  <channelName><!--可选项，xs:string，通道名称--></channelName>
  <deviceID><!--可选项，xs:string，设备 ID--></deviceID>
  <Extensions>
    <serialNumber><!--必选项，xs:string，序列号--></serialNumber>

```

```

    <eventPush><!--必选项, xs:string, 事件或报警推送详情--></eventPush>
  </Extensions>
</EventNotificationAlert>

```

16.4.20 XML_EventNotificationAlert_FastMovingDetectionMsg

关于快速移动报警详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, Ipv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, Ipv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口--></portNo>
  <protocolType><!--可选项, xs:string, 传输通信协议: "HTTP,HTTPS,EHome (设备在
EHOME (ISUP) 协议下上报, protocolType 需要赋值为 EHome) "--></protocolType>
  <macAddress><!--可选项, xs:string, MAC--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
relatedChannelList>
  <dateTime><!--必选项, xs:datetime, 事件触发时间, 如: 2017-04-22T15:39:01+08:00--></
dateTime>
  <activePostCount><!--必选项, xs:integer, 事件触发次数--></activePostCount>
  <eventType><!--必选项, xs:string, "rapidMove"--></eventType>
  <eventState><!--必选项, xs:string, 事件是否有效, 针对持续性事件: "active,inactive"--
></eventState>
  <eventDescription><!--必选项, xs:string, 事件英文描述--></eventDescription>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <DetectionRegionList>
    <DetectionRegionEntry><!--列表-->
      <regionID><!--必选项, xs:string--></regionID>
      <sensitivityLevel><!--可选项, xs:integer, 0..100 --></sensitivityLevel>
      <RegionCoordinatesList><!--可选项, 目标区域-->
        <RegionCoordinates><!--可选项-->
          <positionX><!--必选项, xs:integer, X 轴坐标--></positionX>
          <positionY><!--必选项, xs:integer, Y 轴坐标--></positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
    </DetectionRegionEntry>
  </DetectionRegionList>
  <channelName><!--可选项, xs:string, 通道名称--></channelName>
  <detectionPictureTransType><!--可选项, xs:string, 图片传输方式: "url,binary"--></
detectionPictureTransType>
  <detectionPicturesNumber><!--可选项, xs:integer, 表示布防接收报警中包含的图片数量, 有图
片返回, 没有图片不返回--></detectionPicturesNumber>

```

```

    <bkgUrl><!--可选项, xs:string, 背景侦测图片的 URL, 默认情况下是云存储协议--></bkgUrl>
    <URLCertificationType><!--可选项, xs:string, 图片 url 认证方式: "no"-无 (这个是针对武汉云存储协议), "digest"-摘要认证 (这个针对设备本地存储返回 URL 的方式, 设备例如 NVR/DVR) --></URLCertificationType>
    <pId><!--可选项, xs:string, 图片 ID--></pId>
</EventNotificationAlert>

```

16.4.21 XML_EventNotificationAlert_HDDErrorAlarmMsg

关于硬盘错误检测报警信息的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
    <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
    <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
    <protocolType><!--可选项, xs:string, 传输协议类型: HTTP、HTTPS、EHome--></protocolType>
    <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
    <dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
    <channelID><!--依赖项, xs:string, 通道编号--></channelID>
    <dateTime>
        <!--必选项, xs:datetime, 事件发生时间, ISO 8601 时间格式, 如:
2017-04-22T15:39:01+08:00-->
    </dateTime>
    <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
    <eventType><!--必选项, xs:string, 事件类型, 此处应设为 "diskerror"--></eventType>
    <eventState><!--必选项, xs:string, 事件状态 (持续事件): "active"、"inactive" --></eventState>
    <eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
    <channelName><!--可选项, xs:string, 通道名称--></channelName>
    <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
    <diskNo><!--可选项, xs:integer, 硬盘编号--></diskNo>
    <HDDLlist><!--依赖项, xs:integer, 硬盘列表, 当节点<eventType>值为 "diskfull" 或 "diskerror" 时该节点有效-->
        <HDD>
            <id><!--必选项, xs:integer, 编号, 从 1 开始--></id>
            <diskNumber><!--必选项, xs:integer, 硬盘 ID--></diskNumber>
        </HDD>
    </HDDLlist>
</EventNotificationAlert>

```

16.4.22 XML_EventNotificationAlert_IllegalLoginAlarmMsg

关于非法登录报警信息的 XML 报文

```
<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型:HTTP、HTTPS、EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道编号--></channelID>
  <dateTime>
    <!--必选项, xs:datetime, 事件发生时间, ISO 8601 时间格式, 如:
2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型, 此处应设为"illaccess"--></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): "active"、"inactive" --></
eventState>
  <eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
  <channelName><!--可选项, xs:string, 通道名称--></channelName>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
</EventNotificationAlert>
```

16.4.23 XML_EventNotificationAlert_fielddetection

关于入侵侦测报警详情的 XML 报文

```
<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, Ipv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型:HTTP, HTTPS, EHome--></
protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道 ID--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
  <releatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
releatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, IOS8601 格式的事件发生时间, 例如,
2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 此处事件类型设为 "fielddetection"--></eventType>
```



```

<eventState><!--必选项, xs:string, 事件状态 (持续事件): active, inactive--></eventState>
<eventDescription><!--必选项, xs:string, 事件说明--></eventDescription>
<deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
<DetectionRegionList>
  <DetectionRegionEntry><!--列表-->
    <regionID><!--必选项, xs:string, 侦测区域 ID--></regionID>
    <sensitivityLevel><!--可选项, xs:integer, 灵敏度范围: [0, 100]--></sensitivityLevel>
    <RegionCoordinatesList><!--可选项, 目标区域-->
      <RegionCoordinates><!--可选项-->
        <positionX><!--必选项, xs:integer, x 轴坐标--></positionX>
        <positionY><!--必选项, xs:integer, y 轴坐标--></positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
    <detectionTarget><!--可选项, xs:string, 目标类型: "human"、"vehicle"、"others" --></detectionTarget>
    <TargetRect><!--可选项-->
      <X><!--必选项, xs:float; 目标位置的 x 轴坐标, 范围: [0,1]--></X>
      <Y><!--必选项, xs:float; 目标位置的 y 轴坐标, 范围: [0,1]--></Y>
      <width><!--必选项, xs:float; 目标框宽度, 范围: [0,1]--></width>
      <height><!--必选项, xs:float; 目标框高度, 范围: [0,1]--></height>
    </TargetRect>
  </DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<Extensions>
  <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
  <eventPush><!--必选项, xs:string, 事件或报警推送内容--></eventPush>
</Extensions>
<detectionPictureTransType><!--可选项, xs:string, 图片传输模式: "url" 和 "binary" --></detectionPictureTransType>
<detectionPicturesNumber>
  <!--可选项, xs:integer, 报警图片编号, 如果没有报警图片, 则不返回该节点-->
</detectionPicturesNumber>
<bkgUrl><!--可选项, xs:string, 背景图片 URL--></bkgUrl>
<URLCertificationType>
  <!--可选项, xs:string, 图片 URL 的认证模式: no (无认证), digest (摘要认证, 通过 NVR 或 DVR 返回本地图片 URL) -->
</URLCertificationType>
<pId><!--可选项, xs:string, 图片 ID--></pId>
<visibleLightURL>
  <!--可选项, xs:string, 可见光图片 URL-->
</visibleLightURL>
<thermalURL>
  <!--可选项, xs:string, 热图 URL-->
</thermalURL>
<timestamp><!--可选项, xs:string, 触发报警的帧的时间戳, ISO 8601 时间格式, 例如: 2019-12-30T15:29:23+08:00--></timestamp>

```

```

    <uid><!--可选项, xs:string, 事件上报的唯一标识, 64 字节的长度, 可以使用时间 (精确到毫秒) 加上随即数的方式组成--></uid>
    <isDataRetransmission>
      <!--可选项, boolean, 重传数据标记 (由于网络异常等因素导致的实时检测的数据上传失败后, 设备异常因素恢复后重新上传当时的采集分析数据) -->true
    </isDataRetransmission>
    <ruleID>
      <!--可选项, int, 规则序号; 取值范围: [0, 7]。对应通道视频行为分析规则参数中的规则序号, 对应接口: /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>-->0
    </ruleID>
    <ruleName>
      <!--可选项, string, 规则名称; 长度范围: [0, 32]。对应通道视频行为分析规则参数中的规则名称, 对应接口: /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>-->test
    </ruleName>
  </EventNotificationAlert>

```

16.4.24 XML_EventNotificationAlert_linedetection

关于越界侦测报警详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型: HTTP, HTTPS, EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道 ID--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
  <releatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></releatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, IOS8601 格式的事件发生时间, 例如, 2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 此处事件类型应设为 "linedetection"--></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): "active", "inactive" --></eventState>
  <eventDescription><!--必选项, xs:string, 事件说明--></eventDescription>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <DetectionRegionList>
    <DetectionRegionEntry><!--列表-->

```

```

    <regionID><!--必选项, xs:string, 侦测区域 ID--></regionID>
    <sensitivityLevel><!--可选项, xs:integer, 灵敏度范围: [0, 100]--></
sensitivityLevel>
    <RegionCoordinatesList><!--可选项, 目标区域-->
        <RegionCoordinates><!--可选项-->
            <positionX><!--必选项, xs:integer, x 轴坐标--></positionX>
            <positionY><!--必选项, xs:integer, y 轴坐标--></positionY>
        </RegionCoordinates>
    </RegionCoordinatesList>
    <detectionTarget><!--可选项, xs:string, 目标类型:“human”、“vehicle”、“others” --
></detectionTarget>
    <TargetRect><!--可选项-->
    <X><!--必选项, xs:float; 目标位置的 x 轴坐标, 范围: [0,1]--></X>
    <Y><!--必选项, xs:float; 目标位置的 y 轴坐标, 范围: [0,1]--></Y>
    <width><!--必选项, xs:float; 目标框宽度, 范围: [0,1]--></width>
    <height><!--必选项, xs:float; 目标框高度, 范围: [0,1]--></height>
    </TargetRect>
    </DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<Extensions>
    <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
    <eventPush><!--必选项, xs:string, 事件或报警推送内容--></eventPush>
</Extensions>
    <detectionPictureTransType><!--可选项, xs:string, 图片传输模式:“url” 和“binary” --
></detectionPictureTransType>
    <detectionPicturesNumber>
        <!--可选项, xs:integer, 报警图片编号, 如果没有报警图片, 则不返回该节点-->
    </detectionPicturesNumber>
    <bkgUrl><!--可选项, xs:string, 背景图片 URL--></bkgUrl>
    <URLCertificationType>
        <!--可选项, xs:string, 图片 URL 的认证模式: no (无认证), digest (摘要认证, 通过 NVR 或
DVR 返回本地图片 URL) -->
    </URLCertificationType>
    <pId><!--可选项, xs:string, 图片 ID--></pId>
    <visibleLightURL>
        <!--可选项, xs:string, 可见光图片 URL-->
    </visibleLightURL>
    <thermalURL>
        <!--可选项, xs:string, 热度图 URL-->
    </thermalURL>
    <timeStamp><!--可选项, xs:string, 触发报警的帧的时间戳, ISO 8601 时间格式, 例如:
2019-12-30T15:29:23+08:00--></timeStamp>
    <uid><!--可选项, xs:string, 事件上报的唯一标识, 64 字节的长度, 可以使用时间 (精确到毫秒) 加
上随即数的方式组成--></uid>
    <isDataRetransmission>
        <!--可选项, boolean, 重传数据标记 (由于网络异常等因素导致的实时检测的数据上传失败后, 设备
异常因素恢复后重新上传当时的采集分析数据) -->true
    </isDataRetransmission>

```

```

<ruleID>
  <!--可选项, int, 规则序号;取值范围:[0,7]。对应通道视频行为分析规则参数中的规则序号, 对应
接口:/ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>-->0
</ruleID>
<ruleName>
  <!--可选项, string, 规则名称;长度范围:[0,32]。对应通道视频行为分析规则参数中的规则名称,
对应接口:/ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/
<ruleID>-->test
</ruleName>
</EventNotificationAlert>

```

16.4.25 XML_EventNotificationAlert_LoiteringDetectionAlarmMsg

关于徘徊侦测报警详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 协议类型: HTTP, HTTPS--></protocol>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道 ID--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
  <releatedChannelList><!--可选项, xs:string, 报警关联(同源)通道列表(与 channelID 在同
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
releatedChannelList>
  <dateTime><!--必选项, xs:datetime, ISO 8601 时间格式, 例如,
2017-04-22T15:39:01+08:00--></dateTime>
  <activePostCount><!--必选项, xs:integer--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型为“loitering”--></eventType>
  <eventState><!--必选项, xs:string, 事件状态(持续事件): active-已发生, inactive-未发
生--></eventState>
  <eventDescription><!--必选项, xs:string--></eventDescription>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <DetectionRegionList>
    <DetectionRegionEntry><!--列表-->
      <regionID><!--必选项, xs:string, 侦测区域 ID--></regionID>
      <sensitivityLevel><!--可选项, xs:integer, 灵敏度等级, 其取值范围从 0 到 100--></
sensitivityLevel>
      <RegionCoordinatesList><!--可选项, 目标区域-->
        <RegionCoordinates><!--可选项-->
          <positionX><!--必选项, xs:integer; X 坐标--></positionX>
          <positionY><!--必选项, xs:integer; Y 坐标--></positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>

```

```

</DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<detectionPictureTransType><!--可选项, xs:string, 图片传输方式:"url", "binary" --></detectionPictureTransType>
<detectionPicturesNumber><!--可选项, xs:integer, 报警图片数量, 若无报警图片, 该节点将不返回--></detectionPicturesNumber>
<bkgUrl><!--可选项, xs:string, 背景图片 URL--></bkgUrl>
<URLCertificationType><!--可选项, xs:string, 图片 URL 认证方式:,no-无认证, digest-摘要认证--></URLCertificationType>
<pId><!--可选项, xs:string, 图片 ID--></pId>
<targetAttrs><![CDATA[{
/*可选项*/
  "deviceId": "",
/*可选项, string, 设备 ID, 最大长度为 64 字节*/
  "deviceChannel": ,
/*可选项, integer32, 设备通道号,*/
  "deviceName": "",
/*可选项, string, 设备名称, 最大长度为 128 字节*/
}]]></targetAttrs>
<TaskInfo><!--可选项, 数据中心产品提交的任务信息-->
  <taskID><!--必选项, xs:string, task ID, the maximum size is 64 bytes--></taskID>
  <algorithmType>
    <!--可选项, xs:integer, 算法类型:1-highQualityVehicle (高质量车), 2-highPerformanceVehicle (高性能车), 3-personStructModel (人员结构化建模), 4-vehicleAndPersonStructModel (人和车结构化建模), 5-videoFaceDetect (视频人脸检测), 6-videoFaceDetectModel (视频人脸检测和建模), 7-smart (人脸侦测 (输出坐标框, 没有识别属性)+人体检测 (输出属性, 坐标框)+行为检测) -->
  </algorithmType>
  <taskName><!--可选项, xs:string, 任务名称--></taskName>
  <algorithmID><!--可选项, xs:string, 算法包 ID--></algorithmID>
  <monitorPointID><!--可选项, xs:string, 监控点编号--></monitorPointID>
  <monitorPointName><!--可选项, xs:string, 监控点名称--></monitorPointName>
  <streamType><!--可选项, xs:string, 码流类型:realtime-实时码流, historyvideo-历史视频码流, localvideo-本地视频码流--></streamType>
  <ruleID><!--可选项, xs:string, 规则 ID, 用于区分相同名称的不同任务--></ruleID>
  <ruleCustomName><!--必选项, xs:string, 自定义事件名称, 由调用者生成, 报警时会携带此字段, 长度不超过 128 字节--></ruleCustomName>
  <algorithmInfo><!--可选项, 算法版本信息, 该字段在二次分析时必选-->
    <algorithmId><!--可选项, xs:string, 算法包 ID--></algorithmId>
    <algoVendorId><!--可选项, xs:string, 算法厂商 ID--></algoVendorId>
    <algoVersionId><!--可选项, xs:string, 算法版本 ID--></algoVersionId>
    <algoChipType><!--可选项, xs:string, 芯片类型--></algoChipType>
    <algoModelVersion><!--可选项, xs:string, 模型版本--></algoModelVersion>
  </algorithmInfo>
</TaskInfo>

```

```
<timeStamp><!--可选项, xs:string, 报警被触发时间, 基于 ISO 8601 时间格式--></timeStamp>
</EventNotificationAlert>
```

示例

徘徊侦测报警信息示例报文

```
<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress>10.16.113.5</ipAddress>
  <protocolType>HTTP</protocolType>
  <macAddress>4c:bd:8f:42:eb:d8</macAddress>
  <channelID>1</channelID>
  <dateTime>2018-01-21T12:50:39+08:00</dateTime>
  <activePostCount>117</activePostCount>
  <eventType>loitering</eventType>
  <eventState>active</eventState>
  <eventDescription>loitering alarm</eventDescription>
  <channelName>IPdome</channelName>
</EventNotificationAlert>
```

16.4.26 XML_EventNotificationAlert_MotionDetectionMsg

关于移动侦测事件信息的 XML 报文

```
<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型:HTTP、HTTPS、EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道编号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联(同源)通道列表(与 channelID 在同一
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
relatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, 事件发生时间, ISO 8601 时间格式, 如:
2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型, 此处应设为"VMD"--></eventType>
  <eventState><!--必选项, xs:string, 事件状态(持续事件):"active"、"inactive"--></
eventState>
  <eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
  <DetectionRegionList><!--可选项, 检测区域参数列表(仅限专家模式)-->
```

```

<DetectionRegionEntry><!--列表, 检测区域参数 (仅限专家模式) -->
  <regionID><!--必选项, xs:string, 检测区域 ID--></regionID>
  <sensitivityLevel><!--必选项, xs:integer, 灵敏度, 取值范围: [0,100]--></sensitivityLevel>
  <RegionCoordinatesList><!--必选项, 检测区域坐标-->
    <RegionCoordinates>
      <positionX><!--必选项, xs:integer, X 轴坐标--></positionX>
      <positionY><!--必选项, xs:integer, Y 轴坐标--></positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
<Extensions>
  <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
  <eventPush><!--必选项, xs:string, 事件推送消息--></eventPush>
</Extensions>
<targetType><!--可选项, xs:string, 目标类型: "human" (人), "vehicle" (车),
"human,vehicle" (人+车); 多个目标类型用逗号隔开, 不支持人车分类的移动侦测时, 该节点无须返回--></targetType>
<bkgUrl><!--可选项, xs:string, 背景侦测图片的 URL--></bkgUrl>
<URLCertificationType><!--可选项, xs:string, 图片 url 认证方式: no (无, 针对武汉云存储协议), digest (摘要认证, 针对设备本地存储返回 URL 的方式, 设备例如 NVR/DVR) --></URLCertificationType>
</EventNotificationAlert>

```

16.4.27 XML_EventNotificationAlert_ObjectRemovalDetectionMsg

关于物品拿取报警详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ipAddress><!--依赖项, xs:string, Ipv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, Ipv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口--></portNo>
  <protocolType><!--可选项, xs:string, 传输通信协议: "HTTP,HTTPS,EHome (设备在 EHOME (ISUP) 协议下上报, protocolType 需要赋值为 EHome) "--></protocolType>
  <macAddress><!--可选项, xs:string, MAC--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></relatedChannelList>
  <dateTime><!--必选项, xs:datetime, 事件触发时间, 如: 2017-04-22T15:39:01+08:00--></dateTime>

```

```

<activePostCount><!--必选项, xs:integer, 事件触发次数--></activePostCount>
<eventType><!--必选项, xs:string, "attendedBaggage"--></eventType>
<eventState><!--必选项, xs:string, 事件是否有效, 针对持续性事件: "active,inactive"--></eventState>
<eventDescription><!--必选项, xs:string, 事件英文描述--></eventDescription>
<deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
<DetectionRegionList>
  <DetectionRegionEntry><!--列表-->
    <regionID><!--必选项, xs:string--></regionID>
    <sensitivityLevel><!--可选项, xs:integer, 0..100 --></sensitivityLevel>
    <RegionCoordinatesList><!--可选项, 目标区域-->
      <RegionCoordinates><!--可选项-->
        <positionX><!--必选项, xs:integer, X 轴坐标--></positionX>
        <positionY><!--必选项, xs:integer, Y 轴坐标--></positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<detectionPictureTransType><!--可选项, xs:string, 图片传输方式: "url,binary"--></detectionPictureTransType>
<detectionPicturesNumber><!--可选项, xs:integer, 表示布防接收报警中包含的图片数量, 有图片返回, 没有图片不返回--></detectionPicturesNumber>
<bkgUrl><!--可选项, xs:string, 背景侦测图片的 URL, 默认情况下是云存储协议--></bkgUrl>
<URLCertificationType><!--可选项, xs:string, 图片 url 认证方式: "no"-无 (这个是针对武汉云存储协议), "digest"-摘要认证 (这个针对设备本地存储返回 URL 的方式, 设备例如 NVR/DVR) --></URLCertificationType>
<pId><!--可选项, xs:string, 图片 ID--></pId>
</EventNotificationAlert>

```

16.4.28 XML_EventNotificationAlert_ParkingDetectionAlarmMsg

关于停车侦测报警详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?><EventNotificationAlert version="2.0"
xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 协议类型: HTTP, HTTPS--></protocol>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道 ID--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
  <releatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></releatedChannelList>
  <dateTime><!--必选项, xs:datetime, ISO 8601 时间格式, 例如,

```



```

2017-04-22T15:39:01+08:00--></dateTime>
<activePostCount><!--必选项, xs:integer--></activePostCount>
<eventType><!--必选项, xs:string, 事件类型: 此处需要设为“parking”--></eventType>
<eventState><!--必选项, xs:string, 事件状态(持续事件): active-已发生, inactive-未发生--></eventState>
<eventDescription><!--必选项, xs:string--></eventDescription>
<deviceID><!--可选项, xs:string, 设备ID--></deviceID>
<DetectionRegionList>
  <DetectionRegionEntry><!--列表-->
    <regionID><!--必选项, xs:string, 侦测区域ID--></regionID>
    <sensitivityLevel><!--可选项, xs:integer, 灵敏度等级, 其取值范围从0到100--></sensitivityLevel>
    <RegionCoordinatesList><!--可选项, 目标区域-->
      <RegionCoordinates><!--可选项-->
        <positionX><!--必选项, xs:integer; X坐标--></positionX>
        <positionY><!--必选项, xs:integer; Y坐标--></positionY>
      </RegionCoordinates>
    </RegionCoordinatesList>
  </DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<detectionPictureTransType><!--可选项, xs:string, 图片传输方式: “url”, “binary”--></detectionPictureTransType>
<detectionPicturesNumber><!--可选项, xs:integer, 报警图片数量; 若无报警图片, 该节点将不返回--></detectionPicturesNumber>
<bkgUrl><!--可选项, xs:string, 背景图片URL--></bkgUrl>
<URLCertificationType><!--可选项, xs:string, 图片URL认证方式: no-无(这个是针对武汉云存储协议), digest-摘要认证(这个针对设备本地存储返回URL的方式, 设备例如NVR/DVR)--></URLCertificationType>
<pId><!--可选项, xs:string, 图片ID; detectionPictureTransType 值为binary时表单中图片的Content-ID填入此节点--></pId>
<targetAttrs><!--可选项, 透传字段, 推荐使用以下三个字段-->
  <![CDATA[{
/*可选项, 透传字段, 推荐使用以下三个字段*/
  "deviceId": "1",
/*可选项, 设备ID, string类型, 最大长度为64字节*/
  "deviceChannel": 1,
/*可选项, 设备通道号, integer32类型*/
  "deviceName": "shebei",
/*可选项, 设备名称, string类型, 最大长度为128字节*/
  }]]>
</targetAttrs>
<TaskInfo><!--可选项, 数据中心产品提交的任务信息-->
  <taskID><!--必选项, xs:string, 任务ID, 最大长度为64字节--></taskID>
  <algorithmType>
    <!--可选项, xs:integer, 算法类型: 1-highQualityVehicle(高质量车)、2-highPerformanceVehicle(高性能车)、3-personStructModel(人员结构化建模)、4-vehicleAndPersonStructModel(人和车结构化建模)、5-videoFaceDetect(视频人脸检测)、6-

```

```

videoFaceDetectModel (视频人脸检测和建模)、7-smart (人脸侦测 (输出坐标框, 没有识别属性) + 人体检测 (输出属性, 坐标框) + 行为检测) -->
  </algorithmType>
  <taskName><!--可选项, xs:string, 任务名称--></taskName>
  <algorithmID><!--可选项, xs:string, 算法包 ID--></algorithmID>
  <monitorPointID><!--可选项, xs:string, 监控点编号--></monitorPointID>
  <monitorPointName><!--可选项, xs:string, 监控点名称--></monitorPointName>
  <streamType><!--可选项, xs:string, 流类型: "realtime,historyvideo,localvideo"--></streamType>
  <ruleID><!--可选项, xs:string, 规则 ID, 用于区分相同名称的不同任务; 事件上报中也有此 ruleID, 表示事件和此任务相关--></ruleID>
  <ruleCustomName><!--必选项, xs:string, 用户自定义事件名称; 调用者生成, 报警时会携带此字段, 长度不超过 128 字节--></ruleCustomName>
  <algorithmInfo><!--可选项, 二次分析数据必填, 算法版本信息-->
    <algorithmId><!--可选项, xs:string, 算法包 ID--></algorithmId>
    <algoVendorId><!--可选项, xs:string, 厂商 ID--></algoVendorId>
    <algoVersionId><!--可选项, xs:string, 算法版本号 ID--></algoVersionId>
    <algoChipType><!--可选项, xs:string, 芯片类型--></algoChipType>
    <algoModelVersion><!--可选项, xs:string, 模型版本--></algoModelVersion>
  </algorithmInfo>
</TaskInfo>
  <timeStamp><!--可选项, xs:string, ISO 8601 时间格式, 触发报警的帧的时间戳; 例如: 2019-12-30T15:29:23+08:00--></timeStamp>
</EventNotificationAlert>

```

16.4.29 XML_EventNotificationAlert_PeopleCountingEventMsg

关于客流统计上传事件信息的 XML 报文

```

<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string--></ipAddress>
  <ipv6Address><!--依赖项, xs:string--></ipv6Address>
  <portNo><!--可选项, xs:integer--></portNo>
  <protocolType><!--可选项, xs:string, "HTTP,HTTPS"--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></relatedChannelList>
  <dateTime><!--必选项, xs:datetime, 基于 ISO 8601 标准的时间格式, 如: 2017-04-22T15:39:01+08:00--></dateTime>
  <activePostCount><!--必选项, xs:integer--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型, 此处必须设为 "PeopleCounting" --></eventType>
  <eventState><!--必选项, xs:string, "active,inactive" --></eventState>

```

```

<eventDescription><!--必选项, xs:string--></eventDescription>
<channelName><!--可选项, xs:string--></channelName>
<deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
<peopleCounting>
  <statisticalMethods><!--可选项, xs:string--></statisticalMethods>
  <RealTime>
    <time><!--必选项, xs:string--></time>
  </RealTime>
  <enter><!--可选项, xs:integer--></enter>
  <exit><!--可选项, xs:integer--></exit>
  <pass><!--可选项, xs:integer--></pass>
</peopleCounting>
<childCounting>
  <enter><!--可选项, xs:integer--></enter>
  <exit><!--可选项, xs:integer--></exit>
</childCounting>
</EventNotificationAlert>

```

16.4.30 XML_EventNotificationAlert_PeopleGatheringDetectionAlarmMsg

关于人员聚集侦测报警详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 协议类型: HTTP, HTTPS--></protocol>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道 ID--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
  一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
  relatedChannelList>
  <dateTime><!--必选项, xs:datetime, ISO 8601 时间格式, 例如,
  2017-04-22T15:39:01+08:00--></dateTime>
  <activePostCount><!--必选项, xs:integer--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型: 此处需要设为 "group" --></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): active-已发生, inactive-未发
  生--></eventState>
  <eventDescription><!--必选项, xs:string--></eventDescription>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <DetectionRegionList>
    <DetectionRegionEntry><!--列表-->
      <regionID><!--必选项, xs:string, 侦测区域 ID--></regionID>
      <sensitivityLevel><!--可选项, xs:integer, 灵敏度等级, 其取值范围从 0 到 100--></

```

```

sensitivityLevel>
  <RegionCoordinatesList><!--可选项，目标区域-->
    <RegionCoordinates><!--可选项-->
      <positionX><!--必选项，xs:integer；X坐标--></positionX>
      <positionY><!--必选项，xs:integer；Y坐标--></positionY>
    </RegionCoordinates>
  </RegionCoordinatesList>
</DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项，xs:string，通道名称--></channelName>
<detecationPictureTransType><!--可选项，xs:string，图片传输方式：“url”，“binary” --></
detecationPictureTransType>
  <detecationPicturesNumber><!--可选项，xs:integer，报警图片数量，如果无报警图片，该节点将
不返回--></detecationPicturesNumber>
  <bkgUrl><!--可选项，xs:string，背景图片 URL--></bkgUrl>
  <URLCertificationType><!--可选项，xs:string，图片 URL 认证方式：,no-无认证，digest-摘
要认证--></URLCertificationType>
  <pId><!--可选项，xs:string，图片 ID--></pId>
  <targetAttrs><![CDATA[{
/*可选项*/
  "deviceId": "",
/*可选项，string，设备 ID，最大长度为 64 字节*/
  "deviceChannel": ,
/*可选项，integer32，设备通道号，*/
  "deviceName": "",
/*可选项，string，设备名称，最大长度为 128 字节*/
  }]]></targetAttrs>
  <TaskInfo><!--可选项，数据中心产品提交的任务信息-->
    <taskID><!--必选项，xs:string，task ID，the maximum size is 64 bytes--></
taskID>
    <algorithmType>
      <!--可选项，xs:integer，算法类型：1-hightQualityVehicle（高质量车），2-
hightPerformanceVehicle（高性能车），3-personStructModel（人员结构化建模），4-
vehicleAndPersonStructModel（人和车结构化建模），5-videoFaceDetect（视频人脸检测），6-
videoFaceDetectModel（视频人脸检测和建模），7-smart（人脸侦测（输出坐标框，没有识别属性）+人
体检测（输出属性，坐标框）+行为检测）-->
    </algorithmType>
    <taskName><!--可选项，xs:string，任务名称--></taskName>
    <algorithmID><!--可选项，xs:string，算法包 ID--></algorithmID>
    <monitorPointID><!--可选项，xs:string，监控点编号--></monitorPointID>
    <monitorPointName><!--可选项，xs:string，监控点名称--></monitorPointName>
    <streamType><!--可选项，xs:string，码流类型：realtime-实时码流，historyvideo-历史视
频码流，localvideo-本地视频码流--></streamType>
    <ruleID><!--可选项，xs:string，规则 ID，用于区分相同名称的不同任务--></ruleID>
    <ruleCustomName><!--必选项，xs:string，自定义事件名称，由调用者生成，报警时会携带此字
段，长度不超过 128 字节--></ruleCustomName>
    <algorithmInfo><!--可选项，算法版本信息，该字段在二次分析时必须-->
      <algorithmID><!--可选项，xs:string，算法包 ID--></algorithmID>

```

```

    <algoVendorId><!--可选项, xs:string, 算法厂商 ID--></algoVendorId>
    <algoVersionId><!--可选项, xs:string, 算法版本 ID--></algoVersionId>
    <algoChipType><!--可选项, xs:string, 芯片类型--></algoChipType>
    <algoModelVersion><!--可选项, xs:string, 模型版本--></algoModelVersion>
  </algorithmInfo>
</TaskInfo>
  <timeStamp><!--可选项, xs:string, 报警被触发时间, 基于 ISO 8601 时间格式--></timeStamp>
</EventNotificationAlert>

```

示例

人员聚集报警信息示例报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress>10.16.113.5</ipAddress>
  <protocolType>HTTP</protocolType>
  <macAddress>4c:bd:8f:42:eb:d8</macAddress>
  <channelID>1</channelID>
  <dateTime>2018-01-21T12:50:39+08:00</dateTime>
  <activePostCount>117</activePostCount>
  <eventType>group</eventType>
  <eventState>active</eventState>
  <eventDescription>group alarm</eventDescription>
  <channelName>IPdome</channelName>
  <UID>12345</UID>
</EventNotificationAlert>

```

16.4.31 XML_EventNotificationAlert_PIRAlarmMsg

关于被动红外探测报警信息的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型:HTTP、HTTPS、EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道编号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
relatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, 事件发生时间, ISO 8601 时间格式, 如 :
2017-04-22T15:39:01+08:00-->
  </dateTime>

```

```

<activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
<eventType><!--必选项, xs:string, 事件类型, 此处应设为"PIR"--></eventType>
<eventState><!--必选项, xs:string, 事件状态 (持续事件): "active"、"inactive" --></
eventState>
<eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
</EventNotificationAlert>

```

16.4.32 XML_EventNotificationAlert_regionEntrance

关于进入区域侦测报警详情的 XML 报文

```

<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型: HTTP, HTTPS, EHome--></
protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道 ID--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
relatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, IOS8601 格式的事件发生时间, 例如,
2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 此处事件类型应设为"regionEntrance" --></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): "active", "inactive" --></
eventState>
  <eventDescription><!--必选项, xs:string, 事件说明--></eventDescription>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <DetectionRegionList>
    <DetectionRegionEntry><!--列表-->
      <regionID><!--必选项, xs:string, 侦测区域 ID--></regionID>
      <sensitivityLevel><!--可选项, xs:integer, 灵敏度范围: [0, 100]--></
sensitivityLevel>
      <RegionCoordinatesList><!--可选项, 目标区域-->
        <RegionCoordinates><!--可选项-->
          <positionX><!--必选项, xs:integer, x 轴坐标--></positionX>
          <positionY><!--必选项, xs:integer, y 轴坐标--></positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>

```

```

    <detectionTarget><!--可选项, xs:string, 目标类型:“human”、“vehicle”、“others”-->
</detectionTarget>
    <TargetRect><!--可选项-->
    <X><!--必选项, xs:float;目标位置的 x 轴坐标, 范围:[0, 1]--></X>
    <Y><!--必选项, xs:float; 目标位置的 y 轴坐标, 范围:[0, 1]--></Y>
    <width><!--必选项, xs:float; 目标框宽度, 范围:[0, 1]--></width>
    <height><!--必选项, xs:float; 目标框高度, 范围:[0, 1]--></height>
    </TargetRect>
  </DetectionRegionEntry>
</DetectionRegionList>
  <channelName><!--可选项, xs:string, 通道名称--></channelName>
  <detectionPictureTransType><!--可选项, xs:string, 图片传输模式:“url”和“binary”-->
</detectionPictureTransType>
  <detectionPicturesNumber>
    <!--可选项, xs:integer, 报警图片编号, 如果没有报警图片, 则不返回该节点-->
  </detectionPicturesNumber>
  <bkgUrl><!--可选项, xs:string, 背景图片 URL--></bkgUrl>
  <URLCertificationType>
    <!--可选项, xs:string, 图片 URL 的认证模式:no (无认证), digest (摘要认证, 通过 NVR 或
DVR 返回本地图片 URL) -->
  </URLCertificationType>
  <pId><!--可选项, xs:string, 图片 ID--></pId>
  <visibleLightURL>
    <!--可选项, xs:string, 可见光图片 URL-->
  </visibleLightURL>
  <thermalURL>
    <!--可选项, xs:string, 热度图 URL-->
  </thermalURL>
  <timeStamp><!--可选项, xs:string, 触发报警的帧的时间戳, ISO 8601 时间格式, 例如:
2019-12-30T15:29:23+08:00--></timeStamp>
  <uid><!--可选项, xs:string, 事件上报的唯一标识, 64 字节的长度, 可以使用时间 (精确到毫秒) 加
上随即数的方式组成--></uid>
  <isDataRetransmission>
    <!--可选项, boolean, 重传数据标记 (由于网络异常等因素导致的实时检测的数据上传失败后, 设备
异常因素恢复后重新上传当时的采集分析数据) -->true
  </isDataRetransmission>
  <ruleID>
    <!--可选项, int, 规则序号;取值范围:[0, 7]。对应通道视频行为分析规则参数中的规则序号, 对应
接口:/ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>-->0
  </ruleID>
  <ruleName>
    <!--可选项, string, 规则名称;长度范围:[0, 32]。对应通道视频行为分析规则参数中的规则名称,
对应接口:/ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/
<ruleID>-->test
  </ruleName>
</EventNotificationAlert>

```

16.4.33 XML_EventNotificationAlert_regionExiting

关于离开区域侦测报警详情的 XML 报文

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型: HTTP, HTTPS, EHome--></
protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道 ID--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
relatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, IOS8601 格式的事件发生时间, 例如,
2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 此处事件类型应设为“regionExiting” --></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): “active”, “inactive” --></
eventState>
  <eventDescription><!--必选项, xs:string, 事件说明--></eventDescription>
  <deviceID><!--可选项, xs:string; 设备 ID--></deviceID>
  <DetectionRegionList>
    <DetectionRegionEntry><!--列表-->
      <regionID><!--必选项, xs:string, 侦测区域 ID--></regionID>
      <sensitivityLevel><!--可选项, xs:integer, 灵敏度范围: [0, 100]--></
sensitivityLevel>
      <RegionCoordinatesList><!--可选项, 目标区域-->
        <RegionCoordinates><!--可选项-->
          <positionX><!--必选项, xs:integer, x 轴坐标--></positionX>
          <positionY><!--必选项, xs:integer, y 轴坐标--></positionY>
        </RegionCoordinates>
      </RegionCoordinatesList>
      <detectionTarget><!--可选项, xs:string, 目标类型: “human”、“vehicle”、“others” --
--></detectionTarget>
      <TargetRect><!--可选项-->
        <X><!--必选项, xs:float; 目标位置的 x 轴坐标, 范围: [0,1]--></X>
        <Y><!--必选项, xs:float; 目标位置的 y 轴坐标, 范围: [0,1]--></Y>
        <width><!--必选项, xs:float; 目标框宽度, 范围: [0,1]--></width>
        <height><!--必选项, xs:float; 目标框高度, 范围: [0,1]--></height>
      </TargetRect>
    </DetectionRegionEntry>

```



```

</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<detecationPictureTransType><!--可选项, xs:string, 图片传输模式:“url”和“binary” -->
</detecationPictureTransType>
<detecationPicturesNumber>
  <!--可选项, xs:integer, 报警图片编号, 如果没有报警图片, 则不返回该节点-->
</detecationPicturesNumber>
<bkgUrl><!--可选项, xs:string, 背景图片 URL--></bkgUrl>
<URLCertificationType>
  <!--可选项, xs:string, 图片 URL 的认证模式: no (无认证), digest (摘要认证, 通过 NVR 或
DVR 返回本地图片 URL) -->
</URLCertificationType>
<pId><!--可选项, xs:string, 图片 ID--></pId>
<visibleLightURL>
  <!--可选项, xs:string, 可见光图片 URL-->
</visibleLightURL>
<thermalURL>
  <!--可选项, xs:string, 热度图 URL-->
</thermalURL>
<timeStamp><!--可选项, xs:string, 触发报警的帧的时间戳, ISO 8601 时间格式, 例如:
2019-12-30T15:29:23+08:00--></timeStamp>
<uid><!--可选项, xs:string, 事件上报的唯一标识, 64 字节的长度, 可以使用时间 (精确到毫秒) 加上
随即数的方式组成--></uid>
<isDataRetransmission>
  <!--可选项, boolean, 重传数据标记 (由于网络异常等因素导致的实时检测的数据上传失败后, 设备
异常因素恢复后重新上传当时的采集分析数据) -->true
</isDataRetransmission>
<ruleID>
  <!--可选项, int, 规则序号; 取值范围: [0, 7]。对应通道视频行为分析规则参数中的规则序号, 对应
接口: /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/<ruleID>-->0
</ruleID>
<ruleName>
  <!--可选项, string, 规则名称; 长度范围: [0, 32]。对应通道视频行为分析规则参数中的规则名称,
对应接口: /ISAPI/Intelligent/channels/<channelID>/behaviorRule/<SID>/rule/
<ruleID>-->test
</ruleName>
</EventNotificationAlert>

```

16.4.34 XML_EventNotificationAlert_sceneChangeDetection

关于场景变更侦测报警信息的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>

```

```
<protocolType><!--可选项, xs:string, 传输协议类型:HTTP、HTTPS、EHome--></protocolType>
<macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
<dynChannelID><!--可选项, xs:string, 数字通道编号--></dynChannelID>
<channelID><!--依赖项, xs:string, 通道编号--></channelID>
<relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></relatedChannelList>
<dateTime>
  <!--必选项, xs:datetime, 事件发生时间, ISO 8601 时间格式, 如:
2017-04-22T15:39:01+08:00-->
</dateTime>
<activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
<eventType><!--必选项, xs:string, 事件类型, 此处应设为"scenechangedetection"--></eventType>
<eventState><!--必选项, xs:string, 事件状态 (持续事件): "active"、"inactive" --></eventState>
<eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
<Extensions>
  <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
  <eventPush><!--必选项, xs:string, 事件推送消息--></eventPush>
</Extensions>
</EventNotificationAlert>
```

16.4.35 XML_EventNotificationAlert_TemperatureAlarmMsg

关于温度报警详情的 XML 报文

```
MIME-Version: 1.0
Connection: close
Content-Type: multipart/mixed; boundary=boundary

--boundary
Content-Type: application/xml; charset="UTF-8"
Content-Length: 1645

<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型: "HTTP", "HTTPS", "EHome" --></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道号--></channelID>
  <dateTime>
```

```

    <!--必选项, xs:datetime, 基于 ISO 8601 标准的事件发生时间, 例如:
2017-04-22T15:39:01+08:00-->
    </dateTime>
    <activePostCount><!--必选项, xs:integer, 事件发生时间--></activePostCount>
    <eventType><!--必选项, xs:string, 事件类型, 此处必须设为“TMA”--></eventType>
    <eventState><!--必选项, xs:string, 事件状态 (用于持续事件):“active”, “inactive” --></
eventState>
    <eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
    <DetectionRegionList><!--依赖项-->
    <DetectionRegionEntry><!--必选项-->
    <regionID><!--必选项, xs:string--></regionID>
    <RegionCoordinatesList><!--可选项-->
    <RegionCoordinates><!--可选项-->
    <positionX><!--必选项, xs:integer, X 坐标--></positionX>
    <positionY><!--必选项, xs:integer, Y 坐标--></positionY>
    </RegionCoordinates>
    </RegionCoordinatesList>
    <TMA><!--依赖项-->
    <thermometryUnit><!--必选项, xs:string, 温度单位:“celsius”-摄氏度,
“fahrenheit”-华氏度, “kelvin”-开尔文温度--></thermometryUnit>
    <ruleTemperature><!--必选项, xs:float, 配置的温度阈值---></ruleTemperature>
    <currTemperature><!--必选项, xs:float, 当前温度---></currTemperature>
    <ruleCalibType><!--必选项, xs:string, 规则标定类型:“point”-点, “line”-线, “region”-
框--></ruleCalibType>
    <MaximumTemperaturePoint><!--依赖项, 该节点仅当<ruleCalibType>为“line”或
“region”时才有效-->
    <RegionCoordinates><!--可选项-->
    <positionX><!--必选项, xs:integer, X 坐标--></positionX>
    <positionY><!--必选项, xs:integer, Y 坐标--></positionY>
    </RegionCoordinates>
    </MaximumTemperaturePoint>
    <AbsoluteHigh>
    <elevation><!--可选项, xs:integer, 取值范围为-900 到 2700--></elevation>
    <azimuth><!--可选项, xs:integer, 取值范围为 0 到 3600--></azimuth>
    <absoluteZoom><!--可选项, xs:integer, 取值范围为 0 到 1000--></absoluteZoom>
    </AbsoluteHigh>
    <presetNo><!--可选项, xs:integer, 预置点编号--></presetNo>
    </TMA>
    </DetectionRegionEntry>
    </DetectionRegionList>
    <channelName><!--可选项, xs:string, 通道名称--></channelName>
    <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
    <detectionPicturesNumber><!--可选项, xs:integer--></detectionPicturesNumber>
    <visibleLightURL><!--可选项, xs:string, 可见光图片 URL--></visibleLightURL>
    <thermalURL><!--可选项, xs:string, 热成像图片 URL--></thermalURL>
    <thermalInfoURL><!--可选项, xs:string, 热成像附加信息 URL--></thermalInfoURL>
    <URLCertificationType>
    <!--可选项, xs:string, 图片 URL 认证方式:“no”-无认证, “digest”-摘要认证 (摘要认证用于

```

```

NVR 或 DVR 返回的本地图片 URL) -->
  </URLCertificationType>
</EventNotificationAlert>

--boundary
Content-Disposition: form-data;
Content-Type: image/jpeg
Content-Length: 47432
<Picture data>
.....JFIF.....
...
--boundary--

```

16.4.36 XML_EventNotificationAlert_UnattendedBaggageDetectionAlarmMsg

关于物品遗留报警详情的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, Ipv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, Ipv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备通信端口--></portNo>
  <protocolType><!--可选项, xs:string, 传输通信协议: "HTTP,HTTPS,EHome (设备在
EHOME (ISUP) 协议下上报, protocolType 需要赋值为 EHome) "--></protocolType>
  <macAddress><!--可选项, xs:string, MAC--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
relatedChannelList>
  <dateTime><!--必选项, xs:datetime, 事件触发时间, 如: 2017-04-22T15:39:01+08:00--></
dateTime>
  <activePostCount><!--必选项, xs:integer, 事件触发次数--></activePostCount>
  <eventType><!--必选项, xs:string, "unattendedBaggage"--></eventType>
  <eventState><!--必选项, xs:string, 事件是否有效, 针对持续性事件: "active,inactive"--
></eventState>
  <eventDescription><!--必选项, xs:string, 事件英文描述--></eventDescription>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <DetectionRegionList>
    <DetectionRegionEntry><!--列表-->
      <regionID><!--必选项, xs:string--></regionID>
      <sensitivityLevel><!--可选项, xs:integer, 0..100 --></sensitivityLevel>
      <RegionCoordinatesList><!--可选项, 目标区域-->
        <RegionCoordinates><!--可选项-->
          <positionX><!--必选项, xs:integer, X 轴坐标--></positionX>
          <positionY><!--必选项, xs:integer, Y 轴坐标--></positionY>

```

```

    </RegionCoordinates>
  </RegionCoordinatesList>
</DetectionRegionEntry>
</DetectionRegionList>
<channelName><!--可选项, xs:string, 通道名称--></channelName>
<detectionPictureTransType><!--可选项, xs:string, 图片传输方式: "url,binary"--></
detectionPictureTransType>
<detectionPicturesNumber><!--可选项, xs:integer, 表示布防接收报警中包含的图片数量, 有图
片返回, 没有图片不返回--></detectionPicturesNumber>
<bkgUrl><!--可选项, xs:string, 背景侦测图片的 URL, 默认情况下是云存储协议--></bkgUrl>
<URLCertificationType><!--可选项, xs:string, 图片 url 认证方式: "no"-无 (这个是针对武汉
云存储协议), "digest"-摘要认证 (这个针对设备本地存储返回 URL 的方式, 设备例如 NVR/DVR) --></
URLCertificationType>
<pId><!--可选项, xs:string, 图片 ID--></pId>
</EventNotificationAlert>

```

16.4.37 XML_EventNotificationAlert_VideoLossAlarmMsg

关于视频丢失报警信息的 XML 报文

```

<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型 HTTP、HTTPS、EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道编号--></channelID>
  <dateTime>
    <!--必选项, xs:datetime, 事件发生时间, IOS 8601 时间格式, 如:
2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型, 此处为"videoloss"--></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): "active"、"inactive" --></
eventState>
  <eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
  <channelName><!--可选项, xs:string, 通道名称--></channelName>
  <deviceId><!--可选项, xs:string, 设备 ID--></deviceId>
  <Extensions>
    <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
    <eventPush><!--必选项, xs:string, 事件推送消息--></eventPush>
  </Extensions>
</EventNotificationAlert>

```

16.4.38 XML_EventNotificationAlert_VideoTamperingAlarmMsg

关于视频遮挡侦测报警信息的 XML 报文

```
<?xml version="1.0" encoding="utf-8"?>
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <ipAddress><!--依赖项, xs:string, IPv4 地址--></ipAddress>
  <ipv6Address><!--依赖项, xs:string, IPv6 地址--></ipv6Address>
  <portNo><!--可选项, xs:integer, 设备的通信端口号--></portNo>
  <protocolType><!--可选项, xs:string, 传输协议类型 HTTP, HTTPS, EHome--></protocolType>
  <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
  <dynChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
  <channelID><!--依赖项, xs:string, 通道号--></channelID>
  <relatedChannelList><!--可选项, xs:string, 报警关联 (同源) 通道列表 (与 channelID 在同
  一个相机上的通道), 用于平台收到报警时展示预览或回放, 多个通道号用逗号隔开--></
  relatedChannelList>
  <dateTime>
    <!--必选项, xs:datetime, 事件发生时间, IOS 8601 时间格式, 如
    2017-04-22T15:39:01+08:00-->
  </dateTime>
  <activePostCount><!--必选项, xs:integer, 事件发生次数--></activePostCount>
  <eventType><!--必选项, xs:string, 事件类型, 此处为"shelteralarm"--></eventType>
  <eventState><!--必选项, xs:string, 事件状态 (持续事件): "active"、"inactive" --></
  eventState>
  <eventDescription><!--必选项, xs:string, 事件描述--></eventDescription>
  <channelName><!--可选项, xs:string, 通道名称--></channelName>
  <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
  <Extensions>
    <serialNumber><!--必选项, xs:string, 序列号--></serialNumber>
    <eventPush><!--必选项, xs:string, 事件推送消息--></eventPush>
  </Extensions>
</EventNotificationAlert>
```

16.4.39 XML_ResponseStatus

关于响应状态参数的 XML 报文

```
<?xml version="1.0" encoding="utf-8"?>
<ResponseStatus version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
  <requestURL>
    <!--必选项, 只读, xs:string, 请求的 URL-->
  </requestURL>
  <statusCode>
    <!--必选项, 只读, xs:integer, 状态码: 0, 1-OK (设备正常), 2-Device Busy (设备正忙), 3-
    Device Error (设备错误), 4-Invalid Operation (操作无效), 5-Invalid XML Format (XML 格式
```

```
无效), 6-Invalid XML Content (XML 内容无效), 7-Reboot Required (必须重启), 9-Additional
Error (其余错误) -->
</statusCode>
<statusString>
  <!--必选项, 只读, xs:string, 状态描述: OK (设备正常), Device Busy (设备正忙), Device
Error (设备错误), Invalid Operation (操作无效), Invalid XML Format (XML 格式无效), Invalid
XML Content (XML 内容无效), Reboot (重启), Additional Error (其余错误) -->
</statusString>
<subStatusCode>
  <!--必选项, 只读, xs:string, 错误原因详情描述-->
</subStatusCode>
<MErrCode>
  <!--可选项, xs:string, 按功能模块分类的错误码, 例如, 0x12345678-->
</MErrCode>
<MErrDevSelfEx>
  <!--可选项, xs:string, <MErrCode> 的扩展字段, 用于定义自定义错误码, 且根据功能模块分类-->
</MErrDevSelfEx>
</ResponseStatus>
```

说明

- 关于子状态码与对应错误码的详情, 参见 [ISAPI 返回码](#)。
 - 关于错误码、错误描述与调试建议的详情, 参见 [功能模块化错误码](#)。
-

附录 A. 结构体和枚举

A.1 Data Structure

A.1.1 NET_DVR_TYPE_NAME

关于硬盘录像机类型和名称的结构体

成员	数据类型	描述
dwDVRType	DWORD	硬盘录像机类型。
byDevName	char	硬盘录像机名称。

A.1.2 NET_EHOME_ADB_DEBUG

关于终端 ADB 控制的结构体

成员	数据类型	描述
bAdbEnable	BOOL	是否开启 ADB 控制。
bySecreKey	BYTE[]	密钥。最大长度为 128 字节。

A.1.3 NET_EHOME_ALARM_HOST_LOG

关于报警主机日志查询结果的结构体

成员	数据类型	描述
struLogTime	<u>NET_EHOME_TIME</u>	日志时间信息。
dwMajorType	DWORD	日志主类型：1-报警, 2-异常, 3-操作, 4-事件。 每一种主类型日志包含多种次类型日志, 可参见 <u>报警主机日志类型</u> 。

成员	数据类型	描述
dwMinorType	DWORD	日志次类型，取决于不同的主类型，可参见 报警主机日志类型 。
dwParamType	DWORD	次类型日志的参数类型；大部分表示防区号，如果没有则设为 0。
sUserName	char[]	用户名。最大长度为 32 字节（对应宏定义为“NAME_LEN”）。
sIPAddress	char[]	远程主机的 IP 地址。最大长度为 128 字节。
sLogContext	char[]	日志内容。最大长度为 8×1024 字节（对应宏定义为“MAX_LOG_INFO_LEN”）。
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差（小时），取值为-12, -10, …, +12, +14, 0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差（分钟），取值为-30, 0, 30, 45, 0xff 表示无效。
byRes	BYTE[]	保留，设为 0。最大长度为 62 字节。

A.1.4 NET_EHOME_ALARM_HOST_LOG_COND

关于报警主机日志查询条件的结构体

成员	数据类型	描述
dwMajorType	DWORD	日志主类型：1-报警, 2-异常, 3-操作, 4-事件, 0xffff-所有类型。每一种主类型日志包含多种次类型日志，可参见 报警主机日志类型 。
dwMinorType	DWORD	日志次类型，取决于不同的主类型，可参见 报警主机日志类型 。
struStartTime	<u>NET_EHOME_TIME</u>	开始时间。

成员	数据类型	描述
struStopTime	<u>NET_EHOME_TIME</u>	结束时间。
dwStartIndex	DWORD	查询起始位置，从 0 开始。
dwMaxFileCountPer	DWORD	单次搜索可查询的最大文件数，由实际网络环境决定。建议最大文件数设为 8。
byLocalOrUTC	BYTE	时间类型：0-设备本地时间，即设备 OSD 时间；1-UTC 时间。
byRes	BYTE[]	保留，设为 0。最大长度为 63 字节。

A.1.5 NET_EHOME_ALARM_INFO

报警信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
szAlarmTime	char[]	报警触发时间（本地时间），时间格式为：YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节（对应宏定义为“MAX_TIME_LEN”）。
szDeviceID	char[]	设备 ID。最大长度为 256 字节（对应宏定义为“MAX_DEVICE_ID_LEN”）。
dwAlarmType	DWORD	报警类型，参见枚举 <u>EN_ALARM_TYPE</u> 。
dwAlarmAction	DWORD	报警状态：0-开始，1-停止。
dwVideoChannel	DWORD	视频通道号。
dwAlarmInChannel	DWORD	报警输入号。
dwDiskNumber	DWORD	硬盘号。

成员	数据类型	描述
byRemark	BYTE[]	备注。最大长度为 64 字节（对应宏定义为“MAX_REMARK_LEN”）。
byRetransFlag	BYTE	重传标记：0-实时包，1-重传包。
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差（小时），取值为-12, -10, …, +12, +14, 0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差（分钟），取值为-30, 0, 30, 45, 0xff 表示无效。
byRes1	BYTE	保留。
szAlarmUploadTime	char[]	报警上传时间（本地时间），时间格式为 YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节（对应宏定义为“MAX_TIME_LEN”）。
uStatusUnion	<u>NET_EHOME_ALARM_STATUS_UNION</u>	状态信息。
byRes2	BYTE[]	保留。最大长度为 16 字节。

A.1.6 NET_EHOME_ALARM_ISAPI_INFO

关于通过 HTTP 上传报警信息的结构体。

成员	数据类型	描述
pAlarmData	char*	报警数据。
dwAlarmDataLen	DWORD	报警数据大小。
byDataType	BYTE	数据类型：0-无效，1-XML 格式，2-JSON 格式。
byPicturesNumber	BYTE	图片数量。

成员	数据类型	描述
byRes	BYTE[]	保留。最大长度为 2 字节。
pPicPackData	void*	报警图片数据，参见结构体 <u>NET_EHOME_ALARM_ISAPI_PICDATA</u> 。
byRes1	BYTE[]	保留。最大长度为 32 字节。

A.1.7 NET_EHOME_ALARM_ISAPI_PICDATA

关于通过 HTTP 上传的报警图片数据结构体

成员	数据类型	描述
dwPicLen	DWORD	图片大小。
byRes	BYTE[]	保留。最大长度为 4 字节。
szFilename	char[]	保存图像文件的路径（包括文件名），最大长度为 8256 字节（对应宏定义为“MAX_FILE_PATH_LEN”）。
pPicData	BYTE*	上传的报警图片数据指针。

A.1.8 NET_EHOME_ALARM_LISTEN_PARAM

报警监听参数结构体。

成员	数据类型	描述
struAddress	<u>NET_EHOME_IPADDRES</u>	本地监听信息。IP 为 0.0.0.0 的情况下，默认为本地地址，多个网卡的情况下，默认为从操作系统获取到的第一个地址。
fnMsgCb	<u>EHomeMsgCallBack</u>	[IN]报警回调函数。
pUserData	void*	[IN]用户参数。

成员	数据类型	描述
byProtocolType	BYTE	[IN]协议类型：0-TCP，1-UDP，2-MQTT。
byUseCmsPort	BYTE	[IN]是否复用中心管理服务器（CMS）端口： 0-不复用，非0-复用。 如果复用 CMS，则 byProtocolType 字段无效，报警管理服务器的本地监听地址 struAddress 填本地回环地址，可通过 <u>NET_ECMS_GetSDKLocalCfg</u> 和 <u>NET_ECMS_SetSDKLocalCfg</u> 来获取和配置。
byUseThreadPool	BYTE	[IN]回调报警时，是否使用线程池：0-使用（默认），1-不使用。
byRes1	BYTE	[IN]保留。
dwKeepAliveSec	DWORD	[IN]心跳间隔，单位：s。如果设为 0，表示默认心跳间隔为 30s。
dwTimeOutCount	DWORD	[IN]心跳超时次数。如果设为 0，表示默认次数为 3 次。
byRes	BYTE	[IN]保留，设为 0。最大长度为 20 字节。

相关接口

NET_EALARM_StartListen

A.1.9 NET_EHOME_ALARM_MPDCDATA

关于车载设备 GPS 信息和客流统计信息的结构体

成员	数据类型	描述
byDeviceID	BYTE[]	设备 ID。最大长度为 256 字节。
bySampleTime	char[]	GPS 采样时间，格式：YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节。

成员	数据类型	描述
byTimeZoneIdx	BYTE	时区。
byRetransFlag	BYTE	重传标记：0-实时包，1-重传包。
byRes	BYTE[]	保留。最大长度为 2 字节。
struGpsInfo	<u>NET_EHOME_MPGPS</u>	GPS 信息。
struMPData	<u>NET_EHOME_MPDATA</u>	客流统计信息。

A.1.10 NET_EHOME_ALARM_MSG

报警信息结构体

成员	数据类型	描述
dwAlarmType	DWORD	报警类型。
pAlarmInfo	void*	报警信息详情，不同报警类型对应不同报警详情。
dwAlarmInfoLen	DWORD	保存报警信息详情所需的缓冲区大小。
pXmlBuf	void*	XML 格式的报警信息报文。
dwXmlBufLen	DWORD	保存报警信息 XML 报文所需的缓冲区大小。
sSerialNumber	char[]	设备序列号。最大长度为 12 字节（对应宏定义为“NET_EHOME_SERIAL_LEN”）。
pHttpUrl	void*	URL。
dwHttpUrlLen	DWORD	URL 大小。
byRes	BYTE[]	保留，设为 0。最大长度为 12 字节。

备注

表 A-1 报警类型和详情

dwAlarmType	描述	pAlarmInfo/pXmlBuf
EHOME_ALARM_UNKNOWN	未知报警	无
EHOME_ALARM	基本报警	<u>NET_EHOME_ALARM_INFO</u>
EHOME_ALARM_HEATMAP_REPORT	热度图报警	<u>NET_EHOME_HEATMAP_REPORT</u>
EHOME_ALARM_FACESNAP_REPORT	人脸抓拍报警	<u>NET_EHOME_FACESNAP_REPORT</u>
EHOME_ALARM_GPS	GPS 报警	<u>NET_EHOME_GPS_INFO</u>
EHOME_ALARM_CID_REPORT	报警主机的 CID 报警	<u>NET_EHOME_CID_INFO</u>
EHOME_ALARM_NOTICE_PICURL	图片 URL 报警	<u>NET_EHOME_NOTICE_PICURL</u>
EHOME_ALARM_NOTIFY_FAIL	异步失败通知报警	<u>NET_EHOME_NOTIFY_FAIL_INFO</u>
EHOME_ALARM_SELFDEFINE	自定义报警	自定义报警信息
EHOME_ALARM_DEVICE_NETSWITCH_REPORT	设备网络切换报警	无
EHOME_ALARM_ACS	门禁事件报警	请参考门禁相关 ISUP SDK 开发指南中“ACS”命令的请求报文。
EHOME_ALARM_WIRELESS_INFO	无线网络报警	<u>NET_EHOME_ALARMWIRELESS_INFO</u>
EHOME_ISAPI_ALARM	通过 HTTP 上传的报警	<u>NET_EHOME_ALARM_ISAPI_INFO</u>

dwAlarmType	描述	pAlarmInfo/pXmlBuf
EHOME_INFO_RELEASE_PRIVATE	通过 ISUP 上传的报警	<u>NET_EHOME_TERMINAL_REPORT_INFO</u>
EHOME_ALARM_MPDCDATA	车载客流统计报警	<u>NET_EHOME_ALARM_MPDCDATA</u>

A.1.11 NET_EHOME_ALARM_STATUS_UNION

报警状态联合体

成员	数据类型	描述
byRes	BYTE[]	联合体大小。最大长度为 12 字节。
struDevStatusChanged	<u>NET_EHOME_DEV_STATUS_CHANGED</u>	设备状态改变信息。
struChanStatusChanged	<u>NET_EHOME_CHAN_STATUS_CHANGED</u>	通道状态改变信息。
struHdStatusChanged	<u>NET_EHOME_HD_STATUS_CHANGED</u>	硬盘状态改变信息。
struDevTimeStatus	<u>NET_EHOME_DEV_TIMING_STATUS</u>	定时上传设备状态参数。
struChanTimeStatus	<u>NET_EHOME_CHAN_TIMING_STATUS_SINGLE</u>	定时上传单通道状态参数。
struHdTimeStatus	<u>NET_EHOME_HD_TIMING_STATUS_SINGLE</u>	定时上传硬盘状态参数。

A.1.12 NET_EHOME_ALARM_TIME_CFG

布防计划参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
struSchedTime	<u>NET_EHOME_SCHEDTIME</u> []	布防时间段。最多支持设置 8 个时间段；通过 ISUP SDK 最多只能设置 4 个时间段。最大长度为 8 字节（对应宏定义为“MAX_TIME_SEGMENT”）。
bySchedTimeCount	BYTE	布防计划中的时间段个数，只读。
byRes	BYTE[]	保留。最大长度为 43 字节。

A.1.13 NET_EHOME_ALARM_TIME_COND

关于布防计划配置条件参数的结构体。

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byAlarmType	DWORD	报警类型：0-移动侦测，1-视频丢失，2-遮挡报警，3-报警输入，4-报警输出，9-客流统计。
byWeekday	DWORD	星期：0-星期一，1-星期二，2-星期三，3-星期四，4-星期五，5-星期六，6-星期日。
byRes1	BYTE[]	保留，设为 0。最大长度为 2 字节。
dwChannel	BYTE	通道号，从 1 开始。
byRes2	BYTE[]	保留，设为 0。最大长度为 20 字节。

A.1.14 NET_EHOME_ALARMIN_CFG

报警输入参数结构体。

成员	数据类型	描述
dwSize	DWORD	结构体大小。
sAlarmInName	DWORD[]	报警输入名称。最大长度为 32 字节（对应宏定义为“NAME_LEN”）。
byAlarmInType	BYTE	报警设备状态：0-常开，1-常闭。
byUseAlarmIn	BYTE	是否开启报警输入：0-关闭，1-开启。
byRes1	BYTE[]	保留，设为 0。最大长度为 2 字节。
struLinkageType	<u>NET_EHOME_ALARMIN_LINKAGE_TYPE</u>	报警联动模式。
struRecordChan	<u>NET_EHOME_RECORD_CHAN</u>	关联录像通道。
byRes1	BYTE	保留，设为 0。最大长度为 128 字节。

A.1.15 NET_EHOME_ALARMIN_COND

关于报警输入配置条件参数的结构体。

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwAlarmInNum	DWORD	报警输入号，从 1 开始。
dwPTZChan	DWORD	PTZ 联动视频通道号，从 1 开始。
byRes	BYTE[]	保留，设为 0。最大长度为 20 字节。

A.1.16 NET_EHOME_ALARMIN_LINKAGE_TYPE

报警联动类型结构体

成员	数据类型	描述
byMonitorAlarm	BYTE	报警是否上墙：0-不上墙，1-上墙。
bySound	BYTE	是否开启声音警告：0-不开启，1-开启。
byUpload	BYTE	是否上传报警至中心：0-不上传，1-上传。
byAlarmout	BYTE	是否触发报警输出：0-不触发，1-触发。
byEmail	BYTE	是否开启邮件联动：0-不开启，1-开启。
byRes1	BYTE[]	保留，设为0。最大长度为3字节。
struPTZLinkage	<u>NET_EHOME_LINKAGE</u> <u>_PTZ</u>	PTZ联动参数。
struAlarmOut	<u>NET_EHOME_LINKAGE</u> <u>_ALARMOUT</u>	报警输出联动参数。
byRes	BYTE[]	保留，设为0。最大长度为128字节。

A.1.17 NET_EHOME_ALARMOUT_CFG

报警输出参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
sAlarmOutName	BYTE[]	报警输出名称。最大长度为32字节（对应宏定义为“NAME_LEN”）。
byAlarmInType	WORD	输出延迟：0-5秒，1-10秒，2-30秒，3-1分钟，4-2分钟，5-5分钟，6-10分钟，7-无限延迟。
byRes	BYTE[]	保留，设为0。最大长度为26字节。

A.1.18 NET_EHOME_ALARMOUT_STATUS_CFG

报警输出状态结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byAlarmOutStatus	BYTE	报警输出状态：1-开启，0-关闭。
byRes	BYTE[]	保留。最大长度为 11 字节。

A.1.19 NET_EHOME_ALARMWIRELESSINFO

无线报警信息结构体

成员	数据类型	描述
byDeviceID	BYTE[]	设备 ID。最大长度为 256 字节（对应宏定义为“MAX_DEVICE_ID_LEN”）。
dwDataTraffic	DWORD	已用流量，单位：MB。
bySignalIntensity	BYTE	信号强度，取值范围从 0 到 100。
byRes	BYTE[]	保留。最大长度为 127 字节。

A.1.20 NET_EHOME_AMS_ADDRESS

关于报警管理服务器本地回环地址的结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byEnable	BYTE	是否接收报警：0-关闭 CMS 接收报警功能，1-开启 CMS 接收报警功能。

成员	数据类型	描述
byRes1	BYTE[]	保留，设为 0。最大长度为 3 字节。
struAddress	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	报警管理服务器的本地回环地址信息（如 IP 地址和端口号）。
byRes2	BYTE[]	保留，设为 0。最大长度为 32 字节。

备注

设置报警管理服务器的本地回环地址前，必须先调用 NET_EALARM_Init 和 NET_EALARM_StartListen 开启报警监听。

A.1.21 NET_EHOME_ASYNC_RESP_CB_DATA

回调响应数据结构体

成员	数据类型	描述
pOutBuffer	void*	响应数据详情。
dwOutLen	DWORD	响应数据大小。
dwErrorNo	DWORD	错误码。
dwHandle	DWORD	消息句柄，用来匹配请求和响应。
IUserID	LONG	用户 ID。
byRes	char[]	保留。最大长度为 32 字节。

A.1.22 NET_EHOME_BINARY_RECV_DATA

数据接收参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lpOutBuffer	void*	用于接收数据的缓冲区，数据大小由 <u>NET_ECMS_STDBinaryCfg</u> 的配置命令 (dwCommand) 决定，详见备注。
dwOutBufferSize	DWORD	已接收的数据大小。
byRes	BYTE[]	保留。最大长度为 64 字节。

备注

用于接收数据的缓冲区，数据大小由 NET_ECMS_STDBinaryCfg 的配置命令 (**dwCommand**) 决定：

dwCommand	命令号	lpOutBuffer
NET_EHOME_S2C_TIME_ADJUST	0x9	无
NET_EHOME_S2C_SET_SERVER_INFO	0x18	无
NET_EHOME_S2C_GET_TERMINAL_INFO	0x19	<u>NET_EHOME_TERMINAL_INFO</u>
NET_EHOME_S2C_POST_SCHEDULE	0x50	无
NET_EHOME_S2C_TERMINALCONTROL_V20	0x51	无
NET_EHOME_S2C_TERMINALCONTROL	0x53	无
NET_EHOME_S2C_PLAYCONTROL	0x54	无
NET_EHOME_S2C_SET_IPCINFO	0x55	无
NET_EHOME_S2C_GET_WORKSTATUS	0x56	<u>NET_EHOME_DEV_WORK_STATUS</u>
NET_EHOME_S2C_VERSIONUPGRADE	0x57	无

dwCommand	命令号	IpOutBuffer
NET_EHOME_S2C_SETPLAYERPARAM	0x58	无
NET_EHOME_S2C_SETSERVERADDR	0x5B	无
NET_EHOME_WEATHERINFO	0x61	无
NET_EHOME_S2C_GET_SCREENSHOT	0x62	无
NET_EHOME_S2C_GET_DEFAULTPARAM	0x63	无
NET_EHOME_S2C_REPLACE_MATERIAL	0x64	<u>NET_EHOME_DEV_DEFAULT_PARAM</u>
NET_EHOME_S2C_SET_QUEUE_INFO	0x66	无
NET_EHOME_S2C_PUBLISH_SHOT_PIC	0x67	无
NET_EHOME_S2C_GET_IPCINFO	0x68	<u>NET_EHOME_RELATE_IPC_INFO</u>
NET_EHOME_S2C_SET_TERMINAL_IP	0x69	无
NET_EHOME_S2C_GET_TERMINAL_IP	0x70	<u>NET_EHOME_TERM_IP_ADDRESS</u>
NET_EHOME_S2C_SET_SWITCHPLAN	0x100	无
NET_EHOME_S2C_GET_SWITCHPLAN	0x101	<u>NET_EHOME_SWITCH_PLAN_PARAM</u>
NET_EHOME_S2C_SET_VOLUMEPLAN	0x102	无
NET_EHOME_S2C_GET_VOLUMEPLAN	0x103	<u>NET_EHOME_VOLUME_PLAN_PARAM</u>
NET_EHOME_S2C_SET_CHARACTER	0x106	无
NET_EHOME_S2C_GET_CHARACTER	0x107	<u>NET_EHOME_INSERT_CHARACTER</u>
NET_EHOME_S2C_SET_ADB_CFG	0x110	无
NET_EHOME_S2C_GET_ADB_CFG	0x111	<u>NET_EHOME_ADB_DEBUG</u>
NET_EHOME_S2C_SET_TIEM_ZONE	0x112	无

dwCommand	命令号	lpOutBuffer
NET_EHOME_S2C_GET_TIME_ZONE	0x113	<u>NET_EHOME_TERM_TIME_ZONE</u>
NET_EHOME_S2C_SET_SADP	0x114	无
NET_EHOME_S2C_GET_SADP	0x115	<u>NET_EHOME_DISCOVERY_MODE</u>
NET_EHOME_S2C_GET_COMPONENT	0x116	<u>NET_EHOME_COMPONENT_GROUP_I</u> <u>NFO</u>
NET_EHOME_S2C_TRANS_WITH_RET	0x10A	无
NET_EHOME_S2C_PUBLISH_XML	0x200	无
NET_EHOME_S2C_SCREEN_SHOT	0x201	<u>NET_EHOME_SCREEN_SHOT_RET</u>

A.1.23 NET_EHOME_BINARY_SEND_DATA

数据发送参数结构体。

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lpInBuffer	void*	保存待发送数据的缓冲区，数据大小由 <u>NET_ECMS_STDBinaryCfg</u> 的配置命令 (dwCommand) 决定，详见备注。
dwInBufferSize	DWORD	待发送数据大小。
dwRecvTimeOut	DWORD	接收超时时间。
byRes	BYTE[]	保留。最大长度为 64 字节。

备注

保存待发送数据的缓冲区，数据大小由 NET_ECMS_STDBinaryCfg 的配置命令(**dwCommand**)决定：

dwCommand	命令号	lpInBuffer
NET_EHOME_S2C_TIME_ADJUST	0x9	<u>NET_EHOME_TIME_ADJUST_PARAM</u>
NET_EHOME_S2C_SEND_ERRCODE	0xD	<u>NET_EHOME_ERRCODE_RET</u>
NET_EHOME_S2C_SET_SERVER_INFO	0x18	<u>NET_EHOME_RELEASE_SERVER_INFO</u>
NET_EHOME_S2C_GET_TERMINAL_INFO	0x19	无
NET_EHOME_S2C_LOGOUT	0x20	无
NET_EHOME_S2C_POST_SCHEDULE	0x50	<u>NET_EHOME_POST_SCHEDULE</u>
NET_EHOME_S2C_TERMINALCONTROL_V20	0x51	<u>NET_EHOME_TERMINAL_CONTROL_V20</u>
NET_EHOME_S2C_TERMINALCONTROL	0x53	<u>NET_EHOME_TERMINAL_CONTROL</u>
NET_EHOME_S2C_PLAYCONTROL	0x54	<u>NET_EHOME_TERMINAL_CONTROL</u>
NET_EHOME_S2C_SET_IPCINFO	0x55	<u>NET_EHOME_RELATE_IPC_INFO</u>
NET_EHOME_S2C_GET_WORKSTATUS	0x56	无
NET_EHOME_S2C_VERSIONUPGRADE	0x57	<u>NET_EHOME_UPGRADE_CONN_PARAMETER</u>
NET_EHOME_S2C_SETPLAYERPARAM	0x58	<u>NET_EHOME_PLAYER_PARAM</u>
NET_EHOME_S2C_SETSERVERADDR	0x5B	<u>NET_EHOME_PUBLISH_SERVERADDR</u>
NET_EHOME_WEATHERINFO	0x61	<u>NET_EHOME_WEATHER_INFO</u>
NET_EHOME_S2C_GET_SCREENSHOT	0x62	<u>NET_EHOME_SCREEN_SHOT</u>
NET_EHOME_S2C_GET_DEFAULTPARAM	0x63	无
NET_EHOME_S2C_REPLACE_MATERIAL	0x64	<u>NET_EHOME_REPLACE_MATERIAL</u>

dwCommand	命令号	lpInBuffer
NET_EHOME_S2C_CANCEL_SCHEDULE	0x65	<u>NET_EHOME_POST_SCHEDULE</u>
NET_EHOME_S2C_SET_QUEUE_INFO	0x66	<u>NET_EHOME_QUEUE_INFO</u>
NET_EHOME_S2C_PUBLISH_SHOT_PIC	0x67	<u>NET_EHOME_SHOT_PIC</u>
NET_EHOME_S2C_GET_IPCINFO	0x68	无
NET_EHOME_S2C_SET_TERMINAL_IP	0x69	<u>NET_EHOME_TERM_IP_ADDRESS</u>
NET_EHOME_S2C_GET_TERMINAL_IP	0x70	无
NET_EHOME_S2C_SET_TERMINAL_NAME	0x71	<u>NET_EHOME_TERMINAL_NAME</u>
NET_EHOME_S2C_SET_SWITCHPLAN	0x100	<u>NET_EHOME_SWITCH_PLAN_PARAM</u>
NET_EHOME_S2C_GET_SWITCHPLAN	0x101	无
NET_EHOME_S2C_SET_VOLUMEPLAN	0x102	<u>NET_EHOME_VOLUME_PLAN_PARAM</u>
NET_EHOME_S2C_GET_VOLUMEPLAN	0x103	无
NET_EHOME_S2C_SET_CHARACTER	0x106	<u>NET_EHOME_INSERT_CHARACTER</u>
NET_EHOME_S2C_GET_CHARACTER	0x107	无
NET_EHOME_S2C_RETRANSMIT_TO_TERM	0x108	待传输的数据
NET_EHOME_S2C_SET_ADB_CFG	0x110	<u>NET_EHOME_ADB_DEBUG</u>
NET_EHOME_S2C_GET_ADB_CFG	0x111	无
NET_EHOME_S2C_SET_TIEM_ZONE	0x112	<u>NET_EHOME_TERM_TIME_ZONE</u>
NET_EHOME_S2C_GET_TIME_ZONE	0x113	无
NET_EHOME_S2C_SET_SADP	0x114	<u>NET_EHOME_DISCOVERY_MODE</u>

dwCommand	命令号	lpInBuffer
NET_EHOME_S2C_GET_SADP	0x115	无
NET_EHOME_S2C_GET_COMPONENT	0x116	无
NET_EHOME_S2C_TRANS_WITH_RET	0x10A	待传输的数据
NET_EHOME_S2C_PUBLISH_XML	0x200	<u>NET_EHOME_POST_PUBLISH_XML</u>
NET_EHOME_S2C_SCREEN_SHOT	0x201	<u>NET_EHOME_SCREEN_SHOT_EX</u>

A.1.24 NET_EHOME_BLACKLIST_SEVER

黑名单管理服务器参数结构体

成员	数据类型	描述
struAdd	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	服务器地址。
byServerName	BYTE[]	服务器名称。最大长度为 32 字节。
byUserName	BYTE[]	用户名。最大长度为 32 字节。
byPassWord	BYTE[]	密码。最大长度为 32 字节。
byRes	BYTE[]	保留。最大长度为 64 字节。

A.1.25 NET_EHOME_CHAN_STATUS_CHANGED

通道状态改变参数结构体

成员	数据类型	描述
wChanNO	WORD	通道号
byChanStatus	BYTE	通道状态，按位表示。

成员	数据类型	描述
		bit 0: 0-禁用（模拟通道）/删除（数字通道），1-启用（模拟通道）/添加（数字通道）。 bit 1: 0-不在线，1-在线。 bit 2: 0-无信号，1-有信号。 bit 3: 0-未录像，1-在录像。 bit 4: 0-IP 通道信息未改变，1-IP 通道信息改变。
byRes	BYTE[]	保留。最大长度为 9 字节。

A.1.26 NET_EHOME_CHAN_TIMING_STATUS_SINGLE

单通道状态定时上报参数结构体

成员	数据类型	描述
dwBitRate	DWORD	实际码率，单位：Kbps。
wChanNO	WORD	通道号。
byLinkNum	BYTE	关联的客户端个数。
byRes	BYTE[]	保留。最大长度为 5 字节。

A.1.27 NET_EHOME_CHAR_EFFECT

字符效果结构体

成员	数据类型	描述
dwFontSize	DWORD	字体大小。
dwFontColor	DWORD	字体颜色。
dwBackColor	DWORD	背景颜色。

成员	数据类型	描述
dwBackTransparent	DWORD	背景透明度。
bSubtitlesEnabled	BOOL	是否启用 OSD。
szScrollDirection	BYTE[]	滚动方向：向左，向右，向上，向下。最大长度为 32 字节。
dwScrollSpeed	DWORD	滚动速度。

A.1.28 NET_EHOME_CID_INFO

CID 信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byDeviceID	char[]	设备注册 ID。最大长度为 256 字节（对应宏定义为“MAX_DEVICE_ID_LEN”）。
dwCIDCode	DWORD	CID 码。
dwCIDType	DWORD	CID 报警类型。
dwSubSysNo	DWORD	产生报告的子系统号，0 表示全局报告。子系统范围从 0 到 32。
byCIDDescribe	char[]	CID 报警类型。最大长度为 32 字节（对应宏定义为“CID_DES_LEN”）。
byTriggerTime	char[]	CID 报警触发时间（设备本地时间），格式：YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节（对应宏定义为“MAX_TIME_LEN”）。
byUploadTime	char[]	CID 报警上传时间（设备本地时间），格式：YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。

成员	数据类型	描述
		最大长度为 32 字节（对应宏定义为“MAX_TIME_LEN”）。
struCIDParam	<u>NET_EHOME_CID_PARAM</u>	CID 报警参数。
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差（小时），取值为-12, -10, ..., +12, +14, 0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差（分钟），取值为-30, 0, 30, 45, 0xff 表示无效。
byExtend	BYTE	是否存在扩展字段：0-否，1-是。
byRes1	BYTE[]	保留。最大长度为 5 字节。
pCidInfoEx	void*	只有当 byExtend 设为 1 时，该参数有效，并指向结构体 <u>NET_EHOME_CID_INFO_INTERNAL_EX</u> 。
pPicInfoEx	void*	附加指针信息，指向结构体 <u>NET_EHOME_CID_INFO_PICTUREINFO_EX</u> 。
byRes2	BYTE[]	保留（该成员仅对 32 位操作系统有效）。最大长度为 4 字节。
byRes	BYTE[]	保留。最大长度为 44 字节。

A.1.29 NET_EHOME_CID_INFO_INTERNAL_EX

CID 报警扩展信息结构体

成员	数据类型	描述
byRecheck	BYTE	报警类型：1-视频复核报警，0-普通报警。
byRes	BYTE[]	保留。最大长度为 3 字节。

成员	数据类型	描述
byUUID	char[]	报警 ID。当报警类型为视频复核报警时，该参数应设为 0。最大长度为 64 字节（对应宏定义为“MAX_UUID_LEN”）。
byVideoURL	char[]	视频复核报警的视频 URL，用来从存储服务器中获取视频。只有当 byRecheck 为 1 时，该参数有效。最大长度为 512 字节（对应宏定义为“MAX_URL_LEN”）。
byCIDDescribeEx	char[]	CID 报警扩展信息。最大长度为 256 字节（对应宏定义为“CID_DES_LEN_EX”）。
byVideoType	char[]	视频类型。最大长度为 16 字节（对应宏定义为“MAX_VIDEO_TYPE_LEN”）。
byRes1	BYTE[]	保留。最大长度为 240 字节。

A.1.30 NET_EHOME_CID_INFO_PICTUREINFO_EX

附加图片信息结构体

成员	数据类型	描述
byPictureURL	char[]	图片 URL。最大图片数量为 5（对应宏定义为“MAX_PICTURE_NUM”），URL 最大长度为 512 字节（对应宏定义为“MAX_URL_LEN”）。
byRes1	BYTE[]	保留。最大长度为 512 字节。

A.1.31 NET_EHOME_CID_PARAM

CID 报警信息结构体

成员	数据类型	描述
dwUserType	DWORD	用户类型：1-键盘用户，2-网络用户，其他值表示无效。
IUserNo	LONG	用户编号，当该参数值设为-1时，表示该参数无效。
IZoneNo	LONG	防区号，当该参数值设为-1时，表示该参数无效。
IKeyboardNo	LONG	键盘编号。
IVideoChanNo	LONG	视频通道号。
IDiskNo	LONG	硬盘号。
IModuleAddr	LONG	模块地址。
byUserName	char[]	用户名。最大长度为 32 字节（对应宏定义为“NAME_LEN”）。
byRes	BYTE[]	保留。最大长度为 32 字节。

A.1.32 NET_EHOME_CMS_LISTEN_PARAM

中心管理服务器监听参数结构体

成员	数据类型	描述
struAddress	<u>NET_EHOME_IPADDRES</u>	本地监听信息。IP 为 0.0.0.0 的情况下，默认为本地地址；当存在多个网卡时，默认采用从操作系统获取到的第一个地址。地址格式参见。
fnCB	<u>DEVICE_REGISTER_CB</u>	注册回调函数，参见。
pUserData	void*	用户数据。

成员	数据类型	描述
dwKeepAliveSec	DWORD	心跳间隔 (单位: 秒), 0: 默认为 30 秒。
dwTimeOutCount	DWORD	心跳超时次数 (0: 默认为 3)。
byRes	BYTE[]	保留, 设为 0。最大长度为 24 字节。

相关接口

[NET_ECMS_StartListen](#)

A.1.33 NET_EHOME_CMSCB_DATA

回调数据结构体

成员	数据类型	描述
IUserID	LONG	用户 ID。
dwType	DWORD	回调的数据类型, 详见备注。
dwHandle	DWORD	消息句柄。
bSucc	BOOL	对应同步下的接口返回值 (TRUE-接口返回成功, FALSE-接口返回失败)。
pOutBuffer	void*	设备响应数据, 详见备注。
dwOutLen	DWORD	设备响应数据大小。
dwErrorNo	DWORD	错误码。
byRes	BYTE[]	保留。最大长度为 32 字节。

备注

回调的数据类型

类型宏定义	宏定义值	描述	pOutBuffer
ECMS_CB_TYPE_UNDEFINE	0	未定义	/
ECMS_CB_TYPE_START_GET_REALSTREAM	1	开始预览的请求数据, 对应 <u>NET_ECMS_StartGetRealStream</u> 、 <u>NET_ECMS_StartGetRealStream</u> 接口的异步返回。	指向一个 <u>NET_EHOME_PREVIEWINFO_OUT</u> 结构。
ECMS_CB_TYPE_START_PUSH_REALSTREAM	3	开始传输实时流的请求数据, 对应 <u>NET_ECMS_StartPushRealStream</u> 接口的异步返回。	指向一个 <u>NET_EHOME_PUSHSTREAM_OUT</u> 结构。
ECMS_CB_TYPE_STOP_GET_REALSTREAM_EX	4	停止预览的请求数据, 对应 <u>NET_ECMS_StopGetRealStream</u> 、 <u>NET_ECMS_StopGetRealStreamEx</u> 接口的异步返回。	指向一个 <u>NET_EHOME_STOPSTREAM_PARAM</u> 结构。
ECMS_CB_TYPE_START_PLAYBACK	5	开始回放的请求数据, 对应 <u>NET_ECMS_StartPlayBack</u> 接口的异步返回。	指向一个 <u>NET_EHOME_PLAYBACK_INFO_OUT</u> 结构。
ECMS_CB_TYPE_START_PUSH_PLAYBACK	6	开始传输回放码流的请求数据, 对应 <u>NET_ECMS_StartPushPlayBack</u> 接口的异步返回。	指向一个 <u>NET_EHOME_PUSHPLAYBACK_OUT</u> 结构。
ECMS_CB_TYPE_STOP_PLAYBACK_EX	7	停止回放的请求数据, 对应 <u>NET_ECMS_StopPlayBack</u> 、 <u>NET_ECMS_StopPlayBackEx</u> 接口的异步返回。	指向一个 <u>NET_EHOME_STOPPLAYBACK_PARAM</u> 结构。
ECMS_CB_TYPE_START_VOICETALK_STM	10	通过流媒体服务器开始语音对讲的请求数据, 对应	指向一个 <u>NET_EHOME_VOI</u>

类型宏定义	宏定义 值	描述	pOutBuffer
		<u><i>NET_ECMS_StartVoiceWithStmServer</i></u> 接口的异步返回。	<u><i>CE_TALK_OUT</i></u> 结构。
ECMS_CB_TYPE_START_PUSH_VOICE_STREAM	11	开始传输语音码流请求数据, 对应 <u><i>NET_ECMS_StartPushVoiceStream</i></u> 接口的异步返回。	指向一个 <u><i>NET_EHOME_PUSH_VOICE_OUT</i></u> 结构。
ECMS_CB_TYPE_STOP_VOICE_TALK_STM_EX	12	通过流媒体服务器停止语音对讲的请求数据, 对应 <u><i>NET_ECMS_StopVoiceTalkWithStmServer</i></u> 、 <u><i>NET_ECMS_StopVoiceTalkWithStmServerEx</i></u> 接口的异步返回。	指向一个 <u><i>NET_EHOME_STOP_VOICETALK_STM_PARAM</i></u> 结构。
ECMS_CB_TYPE_ISAPI_PASSTHROUGH	13	透传 ISAPI 的请求数据, 对应 <u><i>NET_ECMS_ISAPIPassThrough</i></u> 接口的异步返回。	指向一个 <u><i>NET_EHOME_PTXML_PARAM</i></u> 结构。
ECMS_CB_TYPE_START_V2_VOICETALK	14	开始语音对讲 (ISUP2.0 协议) 的请求数据, 对应 <u><i>NET_ECMS_StartVoiceTalk</i></u> 接口的异步返回。	指向一个 <u><i>NET_EHOME_VOICETALK_PARAM</i></u> 结构。
ECMS_CB_TYPE_PAUSE_PLAYBACK	15	暂停回放的请求数据, 对应 <u><i>NET_ECMS_PlayBackOperate</i></u> (PLAYBACK_OPERATE_PAUSE) 接口的异步返回。	指向一个 <u><i>NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM</i></u> 结构。
ECMS_CB_TYPE_RESTART_PLAYBACK	16	继续回放的请求数据, 对应 <u><i>NET_ECMS_PlayBackOperate</i></u>	指向一个 <u><i>NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM</i></u>

类型宏定义	宏定义 值	描述	pOutBuffer
		(PLAYBACK_OPERATE_RESTART) 接口的异步返回。	<u>START_PARAM</u> 结构。

相关接口

NET_ECMS_SetCallback

A.1.34 NET_EHOME_COMPONENT_GROUP_INFO

模块分组信息结构体

成员	数据类型	描述
dwComponentNum	DWORD	模块个数。
struSysComponentAdd r	<u>NET_EHOME_SYSCOM PONENT_REDUCED_A DDR</u>	模块信息。
byRes	BYTE[]	保留。最大长度为 20 字节。

A.1.35 NET_EHOME_COMPRESSION_CFG

编码参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byStreamType	BYTE	码流类型：0-视频流，1-视频和音频流。
byPicQuality	BYTE	图像质量：0-最高，1-较高，2-中等，3-低，4-较低，5-最低。
byBitRateType	BYTE	码率类型：0-变码率，1-定码率。

成员	数据类型	描述
byRes1	BYTE	保留，设为 0。
dwResolution	DWORD	分辨率：0-DCIF, 1-CIF, 2-QCIF, 3-4CIF, 4-2CIF, 6-QVGA (320 × 240), 16-VGA, 17-UXGA, 18-SVGA, 19-HD720p, 20-HD900, 21-XVGA, 22-SXGAp (1360 × 1024), 27-1080P (1920 × 1080), 28-2560 × 1920 (500 MP), 29-1600 × 304, 30-2048 × 1536 (300 MP), 31-2448 × 2048 (500 MP), 32-2448 × 1200, 33-2448 × 800, 34-XGA (1024 × 768), 35-SXGA (1280 × 1024), 36-WD1 (960 × 576/960 × 480), 37-HD1080i, 38-WXGA (1440 × 900), 39-HD_F (1920 × 1080/1280 × 720), 40-HD_H (1920 × 540/1280 × 360), 41-HD_Q (960 × 540/630 × 360), 42-2336 × 1744, 43-1920 × 1456, 44-2592 × 2048, 45-3296 × 2472, 46-1376 × 768, 47-1366 × 768, 48-1360 × 768, 49-WSXGA+, 50-720 × 720, 51-1280 × 1280, 52-2048 × 768, 53-2048 × 2048, 54-2560 × 2048, 55-3072 × 2048, 56-2304 × 1296, 57-WXGA (1280 × 800), 58-1600 × 600, 59-2592 × 1944, 60-2752 × 2208, 61-384 × 288, 62-4000 × 3000, 63-4096 × 2160, 64-3840 × 2160, 65-4000 × 2250, 66-3072 × 1728。
dwVideoBitRate	DWORD	视频码率：0-32 Kbps, 1-48 Kbps, 2-64 Kbps, 3-80 Kbps, 4-96 Kbps, 5-128 Kbps, 6-160 Kbps, 7-192 Kbps, 8-224 Kbps, 9-256 Kbps, 10-320 Kbps, 11-384 Kbps, 12-448 Kbps, 13-512 Kbps, 14-640 Kbps, 15-768 Kbps, 16-896 Kbps, 17-1024 Kbps, 18-1280 Kbps, 19-1536

成员	数据类型	描述
		Kbps, 20-1792 Kbps, 21-2048 Kbps, 22-custom。
dwMaxBitRate	DWORD	自定义码率, 只有当 dwVideoBitRate 设为 22 时有效。
dwVideoFrameRate	DWORD	视频帧率: 0-所有帧, 1-1/16x, 2-1/8x, 3-1/4x, 4-1/2x, 5-1x, 6-2x, 7-4x, 8-6x, 9-8x, 10-10x, 11-12x, 12-16x, 13-20x, 14-15x, 15-18x, 16-22x。
wIntervalFrameI	WORD	I 帧间隔, 范围从 1 到 400。
byIntervalBPFrame	BYTE	帧类型: 0-BBP 帧, 1-BP 帧, 2-P 帧。
byRes	BYTE[]	保留, 设为 0。最大长度为 41 字节。

A.1.36 NET_EHOME_COMPRESSION_COND

编码设置条件参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwChannelNum	DWORD	通道号, 从 1 开始。
byCompressionType	BYTE	码流类型: 1-主码流, 2-子码流, 3-第三码流。
byRes	BYTE[]	保留, 设为 0。最大长度为 23 字节。

A.1.37 NET_EHOME_CONFIG

配置参数结构体

成员	数据类型	描述
pCondBuf	void*	条件参数，如通道号；获取参数时，该参数应设为“NULL”。
dwCondSize	DWORD	条件缓冲区大小。
pInBuf	void*	输入参数，获取配置参数时，该参数应设为“NULL”。
dwInSize	DWORD	输入缓冲区大小。
pOutBuf	void*	输出参数，设置参数时，该参数应设为“NULL”。
dwOutSize	DWORD	输出缓冲区大小。
byRes	BYTE[]	保留，设为 0。最大长度为 40 字节。

A.1.38 NET_EHOME_DEV_DEFAULT_PARAM

终端设备默认参数结构体。

成员	数据类型	描述
bAutoLightEnable	BOOL	是否启用自动亮度调节。
dwLightMode	DWORD	亮度值。
dwBackLightLevel	DWORD	背光参数。
dwVolume	DWORD	音量。
bEnableStartLogo	BOOL	是否显示开机 LOGO。
bEnableVolumeTiming	BOOL	是否启用音量定时调节计划。
bEnablePowerTiming	BOOL	是否启用定时开关机计划。

成员	数据类型	描述
bDefaultScheduleEnable	BOOL	是否启用垫片计划。
dwDefaultScheduleId	DWORD	垫片计划 ID。
szDefaultScheduleName	BYTE[]	垫片计划名称。最大长度为 32 字节。
bTemperatureenable	BOOL	是否启用安全温度功能。
dwSecurity	DWORD	温度的安全值。
dwProtectValue	DWORD	温度的保护值。

A.1.39 NET_EHOME_DEV_LOG

终端日志查询结果结构体

成员	数据类型	描述
struLogTime	<i>NET_EHOME_TIME</i>	日志时间。
dwMajorType	DWORD	日志主类型：1-报警，2-异常，3-操作。每一种主类型日志包含多种次类型日志，详细内容可参见 设备日志类型 。
dwMinorType	DWORD	日志次类型，取决于不同的主类型，详细内容可参见 设备日志类型 。
dwParamType	DWORD	日志次类型参数，参见 设备日志类型 。
sLocalUser	char	本地用户名。最大长度为 32 字节（对应宏定义我“NAME_LEN”）。
sRemoteUser	char	远程用户名。最大长度为 32 字节（对应宏定义为“NAME_LEN”）。
sIPAddress	char	远程主机 IP 地址。最大长度为 128 字节。

成员	数据类型	描述
dwChannelNo	DWORD	通道号。
dwHardDiskNo	DWORD	硬盘号。
dwAlarmInputChanNo	DWORD	报警输入通道号。
dwAlarmOutputChanNo	DWORD	报警输出通道号。
sLogContext	char[]	日志信息。最大长度为 8×1024 字节（对应宏定义为“MAX_LOG_INFO_LEN”）。
byTimeDiffH	BYTE	日志时间与国际标准时间（UTC）的小时差，-12 ... +14, 0xff 表示无效
byTimeDiffM	BYTE	日志时间与国际标准时间（UTC）的分钟差，-30, 0, 30, 45, 0xff 表示无效
byRes	BYTE[]	保留，设为 0。最大长度为 62 字节。

A.1.40 NET_EHOME_DEV_LOG_COND

关于设备日志查询条件的结构体。

成员	数据类型	描述
dwMajorType	DWORD	日志主类型：1-报警，2-异常，3-操作，0xffff-所有类型。每一种主类型日志包含多种次类型日志，详细内容可参见 设备日志类型 。
dwMinorType	DWORD	日志次类型，取决于不同的主类型，详细内容可参见 设备日志类型 。
struStartTime	NET_EHOME_TIME	开始时间。
struStopTime	NET_EHOME_TIME	结束时间。
dwStartIndex	DWORD	查询起始位置，从 0 开始。

成员	数据类型	描述
dwMaxFileCountPer	DWORD	单次搜索可查询的最大文件数, 由实际网络环境决定。建议最大文件数为 8。
byLocalOrUTC	BYTE	时间类型: 0-设备本地时间, 即设备 OSD 时间; 1-UTC 时间。
byRes	BYTE[]	保留, 设为 0。最大长度为 63 字节。

A.1.41 NET_EHOME_DEV_REG_INFO

设备注册信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwNetUnitType	DWORD	保留。
byDeviceID	BYTE[]	设备 ID。最大长度为 256 字节 (对应宏定义为 “MAX_DEVICE_ID_LEN”)。
byFirmwareVersion	BYTE[]	固件版本信息。最大长度为 24 字节。
struDevAdd	<u>NET_EHOME_IPADDRESS</u>	设备地址。
dwDevType	DWORD	设备类型。
dwManufacture	DWORD	制造商代码。
byPassWord	BYTE[]	设备登录中心管理服务器的密码, 用户根据需求进行验证。最大长度为 32 字节。
sDeviceSerial	BYTE[]	设备序列号。最大长度为 12 字节 (对应宏定义为 “NET_EHOME_SERIAL_LEN”)。
byReliableTransmission	BYTE	可靠传输。

成员	数据类型	描述
byWebSocketTransmission	BYTE	网络接口传输。
bySupportRedirect	BYTE	是否支持重定向：0-否，1-是。
byDevProtocolVersion	BYTE[]	设备协议版本信息。最大长度为 6 字节。
bySessionKey	BYTE[]	支持 5.0 版本 ISUP 的设备会话密钥。最大长度为 16 字节（对应宏定义为“MAX_MASTER_KEY_LEN”）。
byMarketType	BYTE	注册类型，0-无效（未知类型），1-经销型，2-行业型。
bySupport	BYTE	设备支持的功能情况，从右到左，第一位为 1 表示支持 NPQ 预览（0b00000001），第二位为 1 表示支持 NPQ 回放（0b00000010）。支持“与”操作。
byRes	BYTE[]	保留。最大长度为 25 字节。

A.1.42 NET_EHOME_DEV_REG_INFO_V12

设备注册信息结构体（V12）

成员	数据类型	描述
struRegInfo	<u>NET_EHOME_DEV_REG_INFO</u>	注册信息。
struRegAddr	<u>NET_EHOME_IPADDRESS</u>	设备注册的服务器地址。
sDevName	BYTE[]	设备名称。最大长度为 64 字节（对应宏定义为“MAX_DEVNAME_LEN_EX”）。

成员	数据类型	描述
byDeviceFullSerial	BYTE[]	设备序列号，包含设备型号，时间，编号和扩展信息。最大长度为 64 字节（对应宏定义为“MAX_FULL_SERIAL_NUM_LEN”）。
byRes	BYTE[]	保留。最大长度为 128 字节。

A.1.43 NET_EHOME_DEV_SESSIONKEY

支持 5.0 版本 ISUP 的设备会话密钥信息结构体。

成员	数据类型	描述
sDeviceID	BYTE[]	设备 ID。最大长度为 256 字节（对应宏定义为“MAX_DEVICE_ID_LEN”）。
sSessionKey	BYTE[]	设备的会话密钥。最大长度为 16 字节（对应宏定义为“MAX_EHOME50_MASTER_KEY_LEN”）。

A.1.44 NET_EHOME_DEV_SESSIONKEY_EHOME50

支持 5.0 版本 ISUP 的设备会话密钥参数结构体

成员	数据类型	描述
sDeviceID	BYTE[]	设备 ID。最大长度为 256 字节（对应宏定义为“MAX_DEVICE_ID_LEN”）。
sSessionKey	BYTE[]	设备会话密钥。最大长度为 16 字节（对应宏定义为“MAX_EHOME50_MASTER_KEY_LEN”）。

A.1.45 NET_EHOME_DEV_STATUS_CHANGED

设备状态改变信息结构体

成员	数据类型	描述
byDeviceStatus	BYTE	设备状态：0-正常；1-CPU 使用率超过 85%；2-硬件错误，如串口异常。
byRes	BYTE[]	保留。最大长度为 11 字节。

A.1.46 NET_EHOME_DEV_TIMING_STATUS

关于设备状态定时上传参数的结构体。

成员	数据类型	描述
dwMemoryTotal	DWORD	内存总大小，单位：KB。
dwMemoryUsage	DWORD	内存使用量，单位：KB。
byCPUUsage	BYTE	CPU 使用率（%），范围从 0 到 100。
byMainFrameTemp	BYTE	机箱温度，单位：°C。
byBackPanelTemp	BYTE	背板温度，单位：°C。
byRes	BYTE	保留。

A.1.47 NET_EHOME_DEV_WORK_STATUS

设备工作状态结构体。

成员	数据类型	描述
byIdentifyCode	BYTE[]	终端序列号。最大长度为 32 字节。
dwCpuUsage	DWORD	CPU 使用率。
dwMemTotal	DWORD	内存总大小。
dwMemUsed	DWORD	已使用内存。

成员	数据类型	描述
dwDiskTotal	DWORD	磁盘总容量。
dwDiskUsed	DWORD	磁盘利用率。
dwTemperature	DWORD	终端温度。
bySoftwareVersion	BYTE[]	软件版本号。最大长度为 32 字节。
byDspwareVersion	BYTE[]	DSP 版本号。最大长度为 32 字节。
byHardwareVersion	BYTE[]	硬件版本号。最大长度为 32 字节。
bySystemVersion	BYTE[]	系统版本号。最大长度为 32 字节。
byMacAddress	BYTE[]	物理地址, 最大支持 2 个网卡。每个地址最大长度为 32 字节。
bySwitchEnable	BOOL	是否启用定时开关机计划。
byVolumeEnable	BOOL	是否启用音量定时调节计划。
dwRelateScheNo	DWORD	计划编号。
byScheName	BYTE[]	终端显示的计划编号。最大长度为 100 字节。
byInstallType	BYTE[]	屏幕方向: 竖屏或横屏。最大长度为 32 字节。
byResWidth	DWORD	分辨率 (宽度)。
byResHeight	DWORD	分辨率 (高度)。

A.1.48 NET_EHOME_DEVICE_CFG

设备基本信息结构体。

成员	数据类型	描述
dwSize	DWORD	结构体大小。
sServerName	BYTE[]	设备名称。最大长度为 32 字节 (对应宏定义为 “MAX_DEVICE_NAME_LEN”)。

成员	数据类型	描述
dwServerID	DWORD	设备或遥控器编号, 范围从 0 到 255。
dwRecycleRecord	DWORD	是否支持循环录像: 0-否, 1-是。
dwServerType	DWORD	设备类型: 1-数字视频录像机, 3-数字视频服务器, 30-网络摄像机, 40-网络球机。
dwChannelNum	DWORD	通道个数 (包括模拟和数字通道), 只读。
dwHardDiskNum	DWORD	硬盘个数, 只读。
dwAlarmInNum	DWORD	模拟通道关联的报警输入个数, 只读。
dwAlarmOutNum	DWORD	模拟通道关联的报警输出个数, 只读。
dwRS232Num	DWORD	RS-232 串口个数, 只读。
dwRS485Num	DWORD	RS-485 串口个数, 只读。
dwNetworkPortNum	DWORD	网口个数, 只读。
dwAuxoutNum	DWORD	辅口个数, 只读。
dwAudioNum	DWORD	音频端口个数, 只读。
sSerialNumber	BYTE[]	设备序列号, 只读。最大长度为 128 字节 (对应宏定义为 “MAX_SERIALNO_LEN”)。
dwMajorScale	DWORD	是否支持主口缩放: 0-否, 1-是。
dwMinorScale	DWORD	是否支持辅口缩放: 0-否, 1-是。
byRes	BYTE[]	保留。最大长度为 292 字节。

A.1.49 NET_EHOME_DEVICE_INFO

设备信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwChannelNumber	DWORD	模拟通道个数。
dwChannelAmount	DWORD	通道总数 (包括模拟通道和数字通道)。
dwDevType	DWORD	设备类型：1-数字视频录像机，3-数字视频服务器，30-网络摄像机，40-网络球机。
dwDiskNumber	DWORD	设备中的硬盘数量。
sSerialNumber	BYTE[]	设备序列号。最大长度为 128 字节 (对应宏定义为“MAX_SERIALNO_LEN”)。
dwAlarmInPortNum	DWORD	模拟通道关联的报警输入个数。
dwAlarmInAmount	DWORD	报警输入总数。
dwAlarmOutPortNum	DWORD	模拟通道关联的报警输出个数。
dwAlarmOutAmount	DWORD	报警输出总数。
dwStartChannel	DWORD	起始视频通道号。
dwAudioChanNum	DWORD	语音对讲通道个数。
dwMaxDigitChannelNum	DWORD	设备支持的最大数字通道个数
dwAudioEncType	DWORD	语音对讲的音频格式：0-G.722，1-G.711U，2-G.711A，3-G.726，4-AAC，5-MP2L2。
sSIMCardSN	BYTE[]	SIM 卡序列号 (车载设备扩展)。最大长度为 128 字节 (对应宏定义为“MAX_SERIALNO_LEN”)。
sSIMCardPhoneNum	BYTE[]	SIM 卡手机号码 (车载设备扩展)。最大长度为 32 字节 (对应宏定义为“MAX_PHOMENUM_LEN”)。

成员	数据类型	描述
dwSupportZeroChan	DWORD	支持的零通道个数：0-不支持，1-支持 1 路，2-支持 2 路，以此类推。
dwStartZeroChan	DWORD	零通道的起始编号，默认为 10000。
dwSmartType	DWORD	0-智能（默认），1-专业智能。
wDevClass	WORD	设备类型。
byStartDTalkChan	BYTE	起始数字对讲通道号，区别于模拟对讲通道号，0 表示无数字对讲通道。
byRes	BYTE[]	保留。最大长度为 157 字节。

A.1.50 NET_EHOME_DISCOVERY_MODE

终端 SADP 参数结构体

成员	数据类型	描述
bSadpMode	BOOL	是否开启 SADP。

A.1.51 NET_EHOME_DISPLAY_PARAM

终端显示参数结构体

成员	数据类型	描述
dwR	DWORD	红色。
dwG	DWORD	绿色。
dwB	DWORD	蓝色。
dwOffsetR	DWORD	红色色增量。
dwOffsetG	DWORD	绿色色增量。

成员	数据类型	描述
dwOffsetB	DWORD	蓝色色差量。
dwBrightnessLevel	DWORD	亮度。
dwContrastLevel	DWORD	对比度。
dwSharpnessLevel	DWORD	锐度。
dwSaturationLevel	DWORD	饱和度。
dwHueLevel	DWORD	色调。
bAutoLightEnable	BOOL	是否开启亮度自动调节。
dwLightMode	DWORD	自动调节的亮度值。
dwBackLightLevel	DWORD	背光参数。
bBootLogoEnabled	BOOL	是否显示开机 LOGO。
dwScreenVolume	DWORD	屏幕音量，暂时不支持。
bDefaultScheduleEnable	BOOL	是否启用垫片计划。
dwDefaultScheduleId	DWORD	垫片计划 ID。
byDefaultScheduleName	BYTE[]	垫片计划名称。最大长度为 32 字节。
bTemperaturenable	BOOL	是否启用安全温度功能。
bSecurity	DWORD	温度的安全值。
bProtectValue	DWORD	温度的保护值。

A.1.52 NET_EHOME_ERRCODE_RET

错误码返回结构体

成员	数据类型	描述
dwCmd	DWORD	错误码。
dwErrCode	DWORD	返回错误码的对应名称，参见 <u>NET_EHOME_ERROR_CODE</u> 。

A.1.53 NET_EHOME_ETHERNET

以太网端口信息结构体

成员	数据类型	描述
struDevIP	<u>NET_EHOME_IPADDR</u>	设备 IP 地址。
struDevIPMask	<u>NET_EHOME_IPADDR</u>	掩码地址。
dwNetInterface	DWORD	网卡类型：1-10 MB 半双工，2-10 MB 全双工，3-100 MB 半双工，4-100 MB 全双工，6-1000 MB 全双工，5-10 MB/100 MB/1000 MB 自适应。
wDevPort	WORD	设备端口号，SDK 的默认通信端口号为 8000。
wMTU	WORD	MTU 参数。
byMACAddr	BYTE[]	设备物理地址。最大长度为 6 字节(对应宏定义为“MACADDR_LEN”)。
byRes	BYTE[]	保留。最大长度为 2 字节。

A.1.54 NET_EHOME_FACESNAP_REPORT

人脸抓拍报警信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byDeviceID	char[]	设备注册 ID。最大长度为 256 字节（对应宏定义为“MAX_DEVICE_ID_LEN”）。
dwVideoChannel	DWORD	视频通道号。
byAlarmTime	char[]	报警时间，格式：YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节（对应宏定义为“MAX_TIME_LEN”）。
dwFacePicID	DWORD	人脸图片 ID。
dwFaceScore	DWORD	人脸评分，范围为从 1 到 100。
dwTargetID	DWORD	目标 ID。
struTarketZone	<u>NET_EHOME_ZONE</u>	目标区域的归一化坐标，数值为当前画面尺寸百分比 × 1000，精确到小数点后三位。
struFacePicZone	<u>NET_EHOME_ZONE</u>	人脸子图区域的归一化坐标，数值为当前画面尺寸百分比 × 1000，精确到小数点后三位。
struHumanFeature	<u>NET_EHOME_HUMAN_FEATURE</u>	人体属性。
dwStayDuration	DWORD	画面停留时间，单位：ms。
dwFacePicLen	DWORD	人脸图片大小，单位：字节。
byFacePicUrl	BYTE[]	人脸编号。最大长度为 512 字节（对应宏定义为“MAX_URL_LEN”）。
dwBackgroudPicLen	DWORD	背景图片大小，单位：字节。
byBackgroudPicUrl	BYTE[]	背景图片 URL。最大长度为 512 字节（对应宏定义为“MAX_URL_LEN”）。
byRetransFlag	BYTE	重传标记：0-实时包，1-重传包。

成员	数据类型	描述
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差 (小时), 取值为-12, -10, ..., +12, +14, 0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差 (分钟), 取值为-30, 0, 30, 45, 0xff 表示无效。
byRes	BYTE[]	保留。最大长度为 61 字节。

A.1.55 NET_EHOME_FINDCOND

表 A-2 预览停止请求参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
iChannel	LONG	通道号, 从 1 开始
dwMaxFileCountPer	DWORD	单次搜索最大文件个数, 最大文件个数, 需要确定实际网络环境, 建议最大个数为 8。
struStartTime	<i>NET_EHOME_TIME</i>	开始时间。
struStopTime	<i>NET_EHOME_TIME</i>	结束时间。
enumSearchType	<i>SEARCH_TYPE</i>	查找类型: 0-录像文件查找, 对应 表 16-4 , 1-图片文件查找, 对应 表 16-5 , 2-流量查询, 对应 表 16-6 。
表 16-3	union	查询参数。
byStartIndex	BYTE	询起始位置, 兼容老版本, 新版本中不建议使用。
byRes1	BYTE[]	保留, 最大长度为 3 字节。

成员	数据类型	描述
dwStartIndexEx	DWORD	扩展后的查询起始位置，增加对记录条数过大的支持。
byRes	BYTE[]	保留，最大长度为 124 字节。

表 A-3 unionSearchParam

成员	数据类型	描述
byLen	BYTE[]	联合体大小，最大长度为 64 字节。
<u>表 16-4</u>	struct	录像文件查找。
<u>表 16-5</u>	struct	图片文件查找。
<u>表 16-6</u>	struct	流量查询。

表 A-4 struRecordFileParam

成员	数据类型	描述
dwFileType	DWORD	文件类型 (字节整形数) : 0xff-全部类型录像 (不包含图片类型), 0-定时录像, 1-移动报警, 2-报警触发, 3-报警 动测, 4-报警&动测, 5-命令触发, 6-手动录像, 7-震动报警, 8-环境报警, 9-智能报警 (或者取证录像), 10 (0x0a) -PIR报警, 11 (0x0b) -无线报警, 12 (0x0c) -呼救报警, 13 (0x0d) 全部报警, 100-全部类型图片, 101-车牌识别图片, 102-稽查报警图片, 103 手动抓拍图片, 104-回放抓拍图片。

表 A-5 struPicFileParam

成员	数据类型	描述
dwFileType	DWORD	文件类型：255 (0xff) -全部类型，0 (0x00) -定时抓图，1 (0x01) -移动侦测抓图，2 (0x02) -报警抓图，3 (0x03)-报警 移动侦测抓图，4 (0x04) -报警&移动侦测抓图，5 (0x05) -命令触发抓图，6 (0x06) -手动抓图，7 (0x07)-震动报警抓图，8 (0x08) -环境报警触发抓图，9 (0x09) -智能报警图片，10 (0x0a) -PIR 报警图片，11 (0x0b) -无线报警图片，12 (0x0c) -呼救报警图片，13 (0x0d) -人脸侦测图片，14 (0x0e) -越界侦测图片，15 (0x0f) -入侵区域侦测图片，16 (0x10) -场景变更侦测图片，17 (0x11) -设备本地回放时截图，18 (0x12) -智能侦测图片，19 (0x13) -进入区域侦测图片，20 (0x14) -离开区域侦测图片，21 (0x15)-徘徊侦测图片，22 (0x16) -人员聚集侦测图片，23 (0x17)-快速运动侦测图片，24 (0x18) -停车侦测图片，25 (0x19) -物品遗留侦测图片，26 (0x1a) -物品拿取侦测图片，27 (0x1b) -车牌侦测图

成员	数据类型	描述
		片, 28 (0x1c) -客户端上传图片。

表 A-6 struFlowParam

成员	数据类型	描述
bySearchMode	BYTE	查询模式, 1-按 year 查询, 2-按 month 查询, 3-按 day 查询。

A.1.56 NET_EHOME_FINDDATA

表 A-7 查询录像文件信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
szFileName	char[]	文件名。最大长度为 100 字节 (对应宏定义为 "MAX_FILE_NAME_LEN")。
struStartTime	<u>NET_EHOME_TIME</u>	文件的开始时间。
struStopTime	<u>NET_EHOME_TIME</u>	文件的结束时间。
dwFileSize	DWORD	文件的大小。
dwFileMainType	DWORD	录像文件主类型。
dwFileSubType	DWORD	录像文件子类型。
dwFileIndex	DWORD	录像文件索引。
byRes	BYTE[]	保留, 最大长度为 128 字节。

A.1.57 NET_EHOME_FLOW_COND

流量查询条件结构体

成员	数据类型	描述
bySearchMode	BYTE	查询模式：1-按年查询，2-按月查询，3-按天查询。
byRes	BYTE[]	保留，设为0。最大长度为3字节。
struStartTime	<u>NET_EHOME_TIME</u>	开始时间。
struStopTime	<u>NET_EHOME_TIME</u>	结束时间。
dwStartIndex	DWORD	查询起始位置，从0开始。
dwMaxFileCountPer	DWORD	单次搜索可查询的最大文件数，由实际网络环境决定。建议最大文件数设为8。
byLocalOrUTC	BYTE	时间类型：0-设备本地时间，即设备OSD时间；1-UTC时间。
byRes1	BYTE[]	保留，设为0。最大长度为63字节。

A.1.58 NET_EHOME_FLOW_INFO

流量信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwFlowValue	DWORD	流量值，单位：KB。
dwFlowIndex	DWORD	流量索引号，从0开始。
byRes	BYTE[]	保留，设为0。最大长度为128字节。

A.1.59 NET_EHOME_GPS_CFG

GPS 参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwTransInterval	DWORD	GPS 信息上传时间间隔，单位：秒。
dwMaxSpeed	DWORD	最大速度，单位：cm/h。
dwMinSpeed	DWORD	最小速度，单位：cm/h。
bEnable	BYTE	是否支持：1-否，2-是。
byRes	BYTE[]	保留。最大长度为 63 字节。

A.1.60 NET_EHOME_GPS_INFO

GPS 报警信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
bySampleTime	char[]	GPS 采样时间，格式：YYYY-MM-DD HH:MM:SS。最大取值长度为 32 字节（宏定义：“MAX_DEVICE_ID_LEN”）。
byDeviceID	char[]	设备注册 ID。最大取值长度为 256 字节（宏定义：“MAX_DEVICE_ID_LEN”）。
byDivision	char[]	division[0]的取值：“E”-东经、“W”-西经； division[1]的取值：“N”-北纬、“S”-南纬。最大取值长度为 2 字节。
bySatellites	BYTE	卫星数量。
byPrecision	BYTE	经度因子，其取值等于原始值×100。

成员	数据类型	描述
dwLongitude	DWORD	经度, 取值范围为 0 到 180×3600×100, 转换公式为: 经度=实际度×3600×100+实际分×60×100+实际秒×100。
dwLatitude	DWORD	纬度, 取值范围为 0 到 90×3600×100, 转换公式为: 纬度=实际度×3600×100+实际分×60×100+实际秒×100。
dwDirection	DWORD	方向, 取值范围为 0 到 359.9×100, 0 表示正北方向, 转换公式为: 方向=实际方向×100。
dwSpeed	DWORD	速度, 取值范围为 0 到 999.9×100000, 转换公式为: 速度=实际速度×100000, 单位: 厘米/小时。
dwHeight	DWORD	高度, 单位: 厘米。
byRetransFlag	BYTE	重传标记: 0-实时包、1-重传包。
byLocateMode	BYTE	定位模式: 0-自主定位、1-差分、2-估算、3-数据无效。仅在 NMEA0183 3.00 版本中该参数可用。
byRes1	BYTE[]	保留。最大取值长度为 2 字节。
dwMileage	DWORD	设备里程数统计, 整型, 单位: 米。每天第一次开机或者 00:00:00 时统计结果清零, 后续累加当天里程数上报至中心。
byRes	BYTE[]	保留。最大取值长度为 56 字节。

A.1.61 NET_EHOME_HD_STATUS_CHANGED

硬盘状态改变信息结构体

成员	数据类型	描述
dwVolume	DWORD	硬盘容量，单位：MB。
wHDNo	WORD	硬盘号。
byHDStatus	BYTE	硬盘状态：0-活跃，1-休眠，2-异常，3-休眠 硬盘出错，4-未格式化，5-未连接（网络硬 盘），6-硬盘正在格式化。
byRes	BYTE[]	保留。最大长度为 5 字节。

A.1.62 NET_EHOME_HD_TIMING_STATUS_SINGLE

硬盘状态定时上传参数结构体

成员	数据类型	描述
dwHDFreeSpace	DWORD	硬盘可用容量，单位：MB。
wHDNo	WORD	硬盘号。
byRes	BYTE[]	保留。最大长度为 6 字节。

A.1.63 NET_EHOME_HEATMAP_REPORT

热度图报警信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byDeviceID	char[]	设备注册 ID。最大长度为 256 字节（对应宏 定义为“MAX_DEVICE_ID_LEN”）。
dwVideoChannel	DWORD	视频通道号。

成员	数据类型	描述
byStartTime	char[]	开始时间, 格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。最大长度为 32 字节 (对应宏定义为“MAX_TIME_LEN”)。
byStopTime	char[]	结束时间, 格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。最大长度为 32 字节 (对应宏定义为“MAX_TIME_LEN”)。
struHeatmapValue	<u>NET_EHOME_HEATMAP_VALUE</u>	热度值。
struPixelArraySize	<u>NET_EHOME_PIXEL_ARRAY_SIZE</u>	热度图数据大小。
byPixelArrayData	BYTE[]	热度图数据编号。最大长度为 512 字节。
byRetransFlag	BYTE	重传标记: 0-实时包, 1-重传包。
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差 (小时), 取值为-12, -10, ..., +12, +14, 0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差 (分钟), 取值为-30, 0, 30, 45, 0xff 表示无效。
byRes	BYTE[]	保留。最大长度为 61 字节。

A.1.64 NET_EHOME_HEATMAP_VALUE

热度值信息结构体

成员	数据类型	描述
dwMaxHeatMapValue	DWORD	最大热度值。
dwMinHeatMapValue	DWORD	最小热度值。
dwTimeHeatMapValue	DWORD	平均热度值。

参见

NET_EHOME_HEATMAP_REPORT

A.1.65 NET_EHOME_HTTP_PARAM

关于 HTTP 页面传输的结构体

成员	数据类型	描述
pInBuffer	void*	输入缓冲区 (XML 格式数据)。
dwInSize	DWORD	输入缓冲区大小。
pOutBuffer	void*	输出缓冲区 (XML 格式数据)。
dwOutSize	DWORD	输出缓冲区大小。
dwReturnedXMLLen	DWORD	实际从设备接收到的 XML 数据的长度。
dwRecvTimeOut	DWORD	接受超时, 单位为毫秒。
byRes	BYTE[]	保留。最大长度为 28 字节。

A.1.66 NET_EHOME_HUMAN_FEATURE

人体属性信息结构体

成员	数据类型	描述
byAgeGroup	BYTE	年龄层：1-婴幼儿，2-儿童，3-少年，4-青少年，5-青年，6-壮年，7-中年，8-中老年，9-老年。
bySex	BYTE	性别：1-男，2-女。
byEyeGlass	BYTE	是否戴眼镜：1-否，2-是。
byMask	BYTE	是否戴口罩，1-不戴，2-戴。
byRes	BYTE[]	保留。最大长度为 12 字节。

A.1.67 NET_EHOME_IDENTIFICATION

终端 ID 结构体

成员	数据类型	描述
bySerialNumber	BYTE[]	序列号，目前暂不支持。最大长度为 32 字节。
byIdentifyCode	BYTE[]	终端 ID。最大长度为 32 字节。

A.1.68 NET_EHOME_IMAGE_CFG

图片质量参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byHue	BYTE	色调，取值范围：[0,255]。
byContrast	BYTE	对比度，取值范围：[0,255]。
byBright	BYTE	亮度，取值范围：[0,255]。

成员	数据类型	描述
bySaturation	BYTE	饱和度，取值范围：[0,255]。
byRes	BYTE[]	保留，设为 0。最大长度为 24 字节。

A.1.69 NET_EHOME_INSERT_CHARACTER

关于插播文字信息的结构体

成员	数据类型	描述
dwMessageCnt	DWORD	插播消息内容。
struTextMessage	<u>NET_EHOME_TEXT_MESSAGE</u> []	插播的文字消息。最大长度为 5 字节（对应宏定义为“NET_EHOME_MAX_TEXT_MESSAGE_NUM”）。
szMsgPos	BYTE	文字消息的显示位置：顶部、中间、底部、自定义。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
struPosition	<u>NET_EHOME_POSITION_INFO</u>	文字消息的自定义位置。该成员只在 szMsgPos 的值为自定义时有效。
struCharEffect	<u>NET_EHOME_CHARACTER_EFFECT</u>	字符效果。

A.1.70 NET_EHOME_INSERT_EFFECT

插播效果结构体

成员	数据类型	描述
struCharEffect	<u>NET_EHOME_CHAR_EFFECT</u>	字符效果。
dwPageTime	DWORD	页面显示时间。
dwScrollSpeedWeb	DWORD	画面滚动速度。

参见

NET_EHOME_INSERT_INFO_V20

A.1.71 NET_EHOME_INSERT_INFO

插播参数结构体

成员	数据类型	描述
dwInsertType	DWORD	插播类型，参见枚举 <u>NET_EHOME_PROGRAM_INSERT_TYPE</u> 。
struMaterialInfo	<u>NET_EHOME_INSERT_TEXT_INFO</u>	素材信息。
struProgramInfo	<u>NET_EHOME_INSERT_TEXT_INFO</u>	节目信息。
struPosInfo	<u>NET_EHOME_POSITION_INFO</u>	位置信息。
struInsertEffect	<u>NET_EHOME_INSERT_EFFECT</u>	插播效果。

A.1.72 NET_EHOME_INSERT_INFO_V20

插播参数结构体 (V20)

成员	数据类型	描述
dwInsertType	DWORD	插播类型, 参见枚举 <u>NET_EHOME_PROGRAM_INSERT_TYPE</u> 。
struMaterialInfo	<u>NET_EHOME_INSERT_TEXT_INFO</u>	素材信息。
struProgramInfo	<u>NET_EHOME_INSERT_TEXT_INFO</u>	节目信息。
struPosInfo	<u>NET_EHOME_POSITION_INFO</u>	位置信息。
struInsertEffect	<u>NET_EHOME_INSERT_EFFECT</u>	插播效果。
dwPosMode	DWORD	坐标类型, 参见枚举 <u>NET_EHOME_POS_MODE</u> 。
byRes	BYTE[]	保留。最大长度为 64 字节。

A.1.73 NET_EHOME_INSERT_TEXT_INFO

插播素材或节目信息结构体

成员	数据类型	描述
dwTextNo	DWORD	插播文件序号, 即素材 ID 或节目 ID。
szPlayMode	BYTE[]	播放模式: byTime (按时间插播) 或 byEndTime (按结束时间插播)。最大长度为 32 字节。
dwCountNum	DWORD	播放次数, 暂不支持。
dwPlayDuration	DWORD	播放时间。

成员	数据类型	描述
playEndTime	BYTE[]	按照结束时间进行插播。最大长度为 32 字节。
dwTextSeq	DWORD	用于判断文件 ID 与文件是否相互对应。

A.1.74 NET_EHOME_IPADDR

IP 地址结构体

成员	数据类型	描述
sIpV4	char[]	设备 IPv4 地址。最大长度为 16 字节。
sIpV6	char[]	设备 IPv6 地址。最大长度为 128 字节。

A.1.75 NET_EHOME_IPADDRESS

地址信息结构体

成员	数据类型	描述
szIP	char[]	IP 地址。最大长度为 128 字节。
wPort	WORD	端口号。
byRes	BYTE[]	保留，设为 0。最大长度为 2 字节。

A.1.76 NET_EHOME_IPC_INFO

网络摄像机信息结构体

成员	数据类型	描述
enumIPCType	<u>NET_EHOME_IPC_TYPE</u>	网络摄像机类型。
dwId	DWORD	网络摄像机 ID。
chanNum	DWORD	关联节目的通道号。
enumAddressType	<u>NET_EHOME_ADDRESS_TYPE</u>	地址类型。
szIpcState	BYTE[]	网络摄像机状态：online (在线), offline (离线)。最大长度为 32 字节。
szHostName	BYTE	网络摄像机的域名。最大长度为 32 字节。
szIpVersion	BYTE	IP 地址版本：v4, v6。最大长度为 16 字节。
szIpAddress	BYTE	IPv4 地址。最大长度为 32 字节。
szIpv6Address	BYTE	IPv6 地址。最大长度为 32 字节。
dwPortNo	DWORD	网络摄像机的端口号。
dwIpcChanNum	DWORD	网络摄像机的通道号。
szUserName	BYTE	网络摄像机的用户名。最大长度为 52 字节。
szPassWord	BYTE	网络摄像机的密码。最大长度为 32 字节。
enumTransmitProtocol	<u>NET_EHOME_TRANSMINT_PROTOCOL</u>	传输协议类型。
enumStreamType	<u>NET_EHOME_STREAM_TYPE</u>	码流类型。

A.1.77 NET_EHOME_LINKAGE_ALARMOUT

报警输出联动参数结构体

成员	数据类型	描述
dwAnalogAlarmOutNum	DWORD	模拟报警输出数量，只读。
byAnalogAlarmOut	BYTE[]	模拟报警输出状态，用数组下标表示：0-未启用，1-启用。最大长度为 32 字节（对应宏定义为“MAX_ANALOG_ALARMOUT”）。
byRes	BYTE[]	保留，设为 0。最大长度为 5000 字节。

A.1.78 NET_EHOME_LINKAGE_PTZ

PTZ 联动参数结构体

成员	数据类型	描述
byUsePreset	BYTE	是否调用预置点：0-否，1-是。
byUseCurise	BYTE	是否调用巡航：0-否，1-是。
byUseTrack	BYTE	是否调用轨迹：0-否，1-是。
byRes1	BYTE	保留。
wPresetNo	WORD	预置点编号，取值范围从 1 到 256。有些设备可支持 300 个预置点。
wCuriseNo	WORD	巡航编号，取值范围从 1 到 16。
wTrackNo	WORD	轨迹编号，取值范围从 1 到 16。
byRes2	BYTE[]	保留。最大长度为 6 字节。

A.1.79 NET_EHOME_LISTEN_PREVIEW_CFG

预览监听参数结构体

成员	数据类型	描述
strulIPAdress	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	本地监听信息。IP 为 0.0.0.0 的情况下，默认为本地地址；当存在多个网卡时，默认采用从操作系统获取到的第一个地址。
fnNewLinkCB	<u>PREVIEW_NEWLINK_C</u> <u>B</u>	回调函数，用于接收预览请求的响应报文。
pUser	void*	用户数据。
byLinkMode	BYTE	监听请求的接入方式：0-TCP，1-UDP。
byRes	BYTE[]	保留，设为 0。最大长度为 127 字节。

相关接口

NET_ESTREAM_StartListenPreview

A.1.80 NET_EHOME_LISTEN_VOICETALK_CFG

语音对讲监听参数结构体

成员	数据类型	描述
strulIPAdress	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	本地监听信息。IP 为 0.0.0.0 的情况下，默认为本地地址；当存在多个网卡时，默认采用从操作系统获取到的第一个地址。
fnNewLinkCB	<u>VOICETALK_NEWLINK</u> <u>CB</u>	语音对讲请求的回调函数。
pUserData	void*	用户参数。
byRes	BYTE[]	保留，设为 0。最大长度为 128 字节。

A.1.81 NET_EHOME_LOCAL_ACCESS_SECURITY

设备接入服务器的安全参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byAccessSecurity	BYTE	安全接入级别：0-兼容模式（允许任意 ISUP 协议版本的设备接入），1-普通模式（只支持 ISUP4.0(不含)以下协议版本的设备接入）2-安全模式（只支持 ISUP4.0(含)以上协议版本的设备接入）。
byRes	BYTE[]	保留，设为 0。最大长度为 127 字节。

A.1.82 NET_EHOME_LOCAL_DEV_PINGREQ

关于是否回调心跳的结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byEnablePingReoCallbac k	BYTE	是否回调心跳：0-否（默认）、1-是。
byRes	BYTE[]	保留。最大取值长度为 63 字节。

A.1.83 NET_EHOME_LOCAL_GENERAL_CFG

普通参数结构体

成员	数据类型	描述
byAlarmPictureSeparate	BYTE	是否分开传输报警数据和图片：0-否，1-是。如果分开，回调函数中返回的报警类型为"EHOME_ISAPI_ALARM"。
byRes	BYTE[]	保留，设为0。最大长度为127字节。

A.1.84 NET_EHOME_LOCAL_PLAYBACK_PARAM

表 A-8 回放参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byPlayBackSync	BYTE	是否开启回放同步接收，0-表示使用异步方式，1-表示使用同步方式。
byRes	BYTE[]	保留，最大长度为131字节。

A.1.85 NET_EHOME_LONG_CFG_INPUT

长连接参数结构体

成员	数据类型	描述
fnDataCallBack	<i>LongConfigCallBack</i>	长连接回调函数。
pUser	void*	用户指针。
byRes	BYTE[]	保留。最大长度为32字节。

A.1.86 NET_EHOME_LONG_CFG_SEND

通过长连接发送的数据结构体

成员	数据类型	描述
pDataBuffer	void*	保存待发送数据的缓冲区。
dwDataLen	DWORD	待发送数据的缓冲区大小。
byRes	BYTE[]	保留。最大长度为 32 字节。

A.1.87 NET_EHOME_MAKE_I_FRAME

强制 I 帧参数结构体

成员	数据类型	描述
iChannel	int	通道号。
dwStreamType	DWORD	码流类型：0-主码流，1-子码流，2-保留。
byRes	BYTE[]	保留。最大长度为 40 字节。

A.1.88 NET_EHOME_MANUAL_IOOUT_CTRL

报警输出手动控制参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwChan	DWORD	报警输出号，从 0 开始。
dwDelayTime	DWORD	报警输出持续时间 (s)，值为 0 表示一直输出。
byAction	BYTE	控制类型：0-关闭报警输出，1-开启报警输出。
byRes	BYTE[]	保留，设为 0。最大长度为 19 字节。

A.1.89 NET_EHOME_MPDATA

车载设备客流统计数据结构体

成员	数据类型	描述
byIndex	BYTE	车门编号，从 1 开始。
byVideoChannel	BYTE	视频通道号，从 1 开始。
byRes	BYTE	保留。
byLevel	BYTE	车内拥挤情况：0-空（dwCount<20），1-一般（20<=dwCount<=30），2-较多（30<=dwCount<=50），3-拥挤（dwCcount>=50）
byStarttime	char[]	开始统计时间/开门时间，格式：YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节（对应宏定义为“MAX_TIME_LEN”）。
byStoptime	char[]	停止统计时间/关门时间，格式：YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节（对应宏定义为“MAX_TIME_LEN”）。
dwEnterNum	DWORD	统计时间段内进入人数。
dwLeaveNum	DWORD	统计时间内离开人数。
dwCount	DWORD	当前时刻车内人数。

A.1.90 NET_EHOME_MPGPS

车载客流统计的 GPS 定位信息结构体

成员	数据类型	描述
iLongitude	LONG	经度信息。
iLatitude	LONG	纬度信息。

成员	数据类型	描述
iSpeed	LONG	速度。
iDirection	LONG	方向。

A.1.91 NET_EHOME_NETWORK_CFG

网络参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
struEtherNet	<u>NET_EHOME_ETHERNET</u>	以太网口信息。
struGateWayIP	<u>NET_EHOME_IPADDR</u>	网关地址。
struMultiCastIP	<u>NET_EHOME_IPADDR</u>	多播地址。
struDDNSServer1IP	<u>NET_EHOME_IPADDR</u>	DDNS1 的 IP 地址。
struDDNSServer2IP	<u>NET_EHOME_IPADDR</u>	DDNS2 的 IP 地址。
struAlarmHostIP	<u>NET_EHOME_IPADDR</u>	报警主机 IP 地址。
wAlarmHostPort	WORD	报警主机端口号。
wIPResolverPort	WORD	解析服务器端口号。
struIPResolver	<u>NET_EHOME_IPADDR</u>	解析服务器地址。
struPPPoE	<u>NET_EHOME_PPPOECF</u>	PPPoE 参数。
wHTTPPort	WORD	HTTP 通信端口号。
byRes	BYTE[]	保留。最大长度为 674 字节。

A.1.92 NET_EHOME_NEWLINK_CB_MSG

预览请求参数结构体 (适用于 64 位 Windows 或 Linux 操作系统)

成员	数据类型	描述
szDeviceID	BYTE[]	设备 ID。最大长度为 256 字节 (对应宏定义为 “MAX_DEVICE_ID_LEN”)。
iSessionID	LONG	设备取流会话 ID。
dwChannelNo	DWORD	通道号。
byStreamType	BYTE	码流类型：0-主码流，1-子码流。
byRes1	BYTE[]	保留，最大长度为 2 字节。
byStreamFormat	BYTE	码流封装格式：0-PS，1-标准流格式。
sDeviceSerial	char[]	设备序列号。最大长度为 12 字节 (对应宏定义为 “NET_EHOME_SERIAL_LEN”)。
fnPreviewDataCB	<i>PREVIEW_DATA_CB</i>	数据回调函数，其大小为 8 字节。
pUserData	void*	用户参数，其大小为 8 字节。
byRes	BYTE[]	保留，设为 0。最大长度为 96 字节。

预览请求参数结构体 (适用于 32 位 Windows 或 Linux 操作系统)

成员	数据类型	描述
szDeviceID	BYTE[]	设备 ID。最大长度为 256 字节 (对应宏定义为 “MAX_DEVICE_ID_LEN”)。
iSessionID	LONG	设备取流会话 ID。
dwChannelNo	DWORD	通道号。
byStreamType	BYTE	码流类型：0-主码流，1-子码流。
byRes1	BYTE[]	保留，最大长度为 2 字节。
byStreamFormat	BYTE	码流封装格式：0-PS，1-标准流格式。

成员	数据类型	描述
sDeviceSerial	char[]	设备序列号。最大长度为 12 字节（对应宏定义为“NET_EHOME_SERIAL_LEN”）。
fnPreviewDataCB	<i>PREVIEW_DATA_CB</i>	数据回调函数，其大小为 4 字节。
byRes2	BYTE[]	保留，最大长度为 4 字节。
pUserData	void*	用户参数，其大小为 4 字节。
byRes3	BYTE[]	保留，最大长度为 4 字节。
byRes	BYTE[]	保留，设为 0。最大长度为 96 字节。

A.1.93 NET_EHOME_NOTICE_PICURL

图片 URL 信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byDeviceID	char[]	设备注册 ID。最大长度为 256 字节（对应宏定义为“MAX_DEVICE_ID_LEN”）。
wPicType	WORD	图片类型：0-定时抓图上传中心，1-报警抓图上传中心，2-手动抓图上传中心，3-下载图片，4-外部设备触发抓图上传中心。
wAlarmType	WORD	报警类型：6-报警输入，7-视频遮挡报警，8-移动侦测报警，11-PIR 报警，12-门磁报警，3-呼救报警。只有当 wPicType 设为 1 时该参数有效。
dwAlarmChan	DWORD	报警通道号，只有当 wPicType 设为 1 时，该参数有效。

成员	数据类型	描述
byAlarmTime	char[]	报警触发时间, 格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。只有当 wPicType 设为 1 时该参数有效。最大长度为 32 字节 (对应宏定义为 "MAX_TIME_LEN")。
dwCaptureChan	DWORD	抓拍通道号。
byPicTime	char[]	图片显示时间, 格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。最大长度为 32 字节 (对应宏定义为 "MAX_TIME_LEN")。
byPicUrl	char[]	图片 URL。最大长度为 512 字节 (对应宏定义为 "MAX_URL_LEN")。
dwManualSnapSeq	DWORD	请求序列号, 只有当 PicType 设为 2 或 3 时, 该参数有效。
byRetransFlag	BYTE	重传标记: 0-实时包, 1-重传包。
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差 (小时), 取值为 -12, -10, ..., +12, +14, 0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差 (分钟), 取值为 -30, 0, 30, 45, 0xff 表示无效。
byRes	BYTE[]	保留。最大长度为 29 字节。

A.1.94 NET_EHOME_NOTIFY_FAIL_INFO

异步失败通知信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byDeviceID	char[]	设备注册 ID。最大取值长度为 256 字节（宏定义：“MAX_DEVICE_ID_LEN”）。
wFailedCommand	WORD	失败的命令：1-抓图失败。
wPicType	WORD	图片类型：0-定时抓图并上传至中心、1-报警触发抓图并上传至中心、2-手动抓图并上传至中心、3-下载图片、4-外部设备触发抓图并上传至中心。仅当 wFailedCommand 设为“1”时该参数有效。
dwManualSnapSeq	DWORD	请求流水号，当 PicType 为 2 或 3 时有效。
byRetransFlag	BYTE	重传标志位：0-实时包、1-重传包。
byRes	BYTE[]	保留。最大取值长度为 31 字节。

A.1.95 NET_EHOME_PASSTHROUGH_PARAM

协议参数结构体

成员	数据类型	描述
dwSequence	DWORD	报文序号。
dwUUID	DWORD	会话 ID。
byRes	BYTE[]	保留。最大长度为 64 字节。

A.1.96 NET_EHOME_PASSWORD_CFG

终端密码参数结构体

成员	数据类型	描述
szOldPwd	BYTE[]	旧密码。最大长度为 32 字节。
szNewPwd	BYTE[]	新密码。最大长度为 32 字节。

A.1.97 NET_EHOME_PIC_CFG

OSD 显示参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byChannelName	BYTE[]	通道名称。最大长度为 32 字节（对应宏定义为“NAME_LEN”）。
bIsShowChanName	BYTE	是否显示通道名称：0-否，1-是。
wChanNameXPos	WORD	通道名称显示的 X 轴坐标，按照 704 × 576 来配置，坐标值为 16 的倍数。
wChanNameYPos	WORD	通道名称显示的 Y 轴坐标，按照 704 × 576 来配置，坐标值为 16 的倍数。
bIsShowOSD	BYTE	是否显示日期信息：0-否，1-是。
wOSDXPos	WORD	OSD 显示的 X 轴坐标，按照 704 × 576 来配置，坐标值为 16 的倍数。
wOSDYPos	WORD	OSD 显示的 Y 轴坐标，按照 704 × 576 来配置，坐标值为 16 的倍数。
byOSDType	WORD	OSD 格式：年/月/日，0：XXXX-XX-XX(年-月-日)，1：XX-XX-XXXX(月-日-年)，2：XXXX 年 XX 月 XX 日，3：XX 月 XX 日 XXXX 年，4：XX-XX-XXXX(日-月-年)，5：XX 日 XX 月 XXXX 年

成员	数据类型	描述
byOSDAtrib	BYTE	OSD 属性：0：不显示 OSD，1：透明，闪烁，2：透明，不闪烁，3：闪烁，不透明，4：不透明，不闪烁
byRes1	BYTE[]	保留，设为 0。最大长度为 2 字节。
bIsShowWeek	BYTE	是否显示星期：0-否，1-是。
byRes	BYTE[]	保留，设为 0。最大长度为 64 字节。

A.1.98 NET_EHOME_PIC_FILE

图片文件查找结果结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
szFileName	char[]	图片文件名。最大长度为 100 字节(对应宏定义为“MAX_FILE_NAME_LEN”)。
struPicTime	<u>NET_EHOME_TIME</u>	图片生成时间。
dwFileSize	DWORD	图片文件大小。
dwFileMainType	DWORD	抓图类型：0xff-全部类型，0 (0x00) -定时抓图，1 (0x01) -移动侦测抓图，2 (0x02) -报警抓图，3 (0x03) -报警或移动侦测抓图，4 (0x04) -报警和移动侦测抓图，5 (0x05) -命令触发抓图，6 (0x06) -手动抓图，7 (0x07) -震动报警抓图，8 (0x08) -环境报警触发抓图，9 (0x09) -智能报警抓图，10 (0x0a) -PIR 报警抓图，11 (0x0b) -无线报警抓图，12 (0x0c) -呼救报警抓图，13 (0x0d) -人脸侦测抓图，14 (0x0e) -越界侦测抓图，15 (0x0f)-

成员	数据类型	描述
		入侵区域侦测抓图, 16 (0x10) -场景变更侦测抓图, 17 (0x11) -设备本地回放截图, 18 (0x12) -智能侦测抓图, 19 (0x13) -进入区域侦测抓图, 20 (0x14) -离开区域侦测抓图, 21 (0x15) -徘徊侦测抓图, 22 (0x16) -人员聚集侦测抓图, 23 (0x17) -快速移动侦测抓图, 24 (0x18) -停车侦测抓图, 25 (0x19) -物品遗留侦测抓图, 26 (0x1a) -物品拿取侦测抓图, 27 (0x1b) -车牌侦测抓图, 28 (0x1c) -客户端上传图片
dwFileIndex	DWORD	图片文件编号, 从 0 开始。
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差 (小时), 取值为-12, -10, ..., +12, +14, 0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差 (分钟), 取值为-30, 0, 30, 45, 0xff 表示无效。
byRes	BYTE[]	保留, 设为 0。最大长度为 126 字节。

A.1.99 NET_EHOME_PIC_FILE_COND

图片文件查找条件结构体

成员	数据类型	描述
dwChannel	DWORD	通道号。
dwPicType	DWORD	抓图类型 : 0xff-全部类型, 0 (0x00) -定时抓图, 1 (0x01) -移动侦测抓图, 2 (0x02) -报警抓图, 3 (0x03) -报警或移动侦测抓图, 4 (0x04) -报警和移动侦测抓图, 5 (0x05) -命令触发抓图, 6 (0x06) -手动抓图, 7 (0x07)-

成员	数据类型	描述
		震动报警抓图, 8 (0x08) -环境报警触发抓图, 9 (0x09) -智能报警抓图, 10 (0x0a) -PIR 报警抓图, 11 (0x0b) -无线报警抓图, 12 (0x0c) -呼救报警抓图, 13 (0x0d) -人脸侦测抓图, 14 (0x0e) -越界侦测抓图, 15 (0x0f) -入侵区域侦测抓图, 16 (0x10) -场景变更侦测抓图, 17 (0x11) -设备本地回放截图, 18 (0x12) -智能侦测抓图, 19 (0x13) -进入区域侦测抓图, 20 (0x14) -离开区域侦测抓图, 21 (0x15) -徘徊侦测抓图, 22 (0x16) -人员聚集侦测抓图, 23 (0x17) -快速移动侦测抓图, 24 (0x18) -停车侦测抓图, 25 (0x19) -物品遗留侦测抓图, 26 (0x1a) -物品拿取侦测抓图, 27 (0x1b) -车牌侦测抓图, 28 (0x1c) -客户端上传图片
struStartTime	<i>NET_EHOME_TIME</i>	开始时间。
struStopTime	<i>NET_EHOME_TIME</i>	结束时间。
dwStartIndex	DWORD	查询起始位置, 从 0 开始。
dwMaxFileCountPer	DWORD	单次搜索可查询的最大文件数, 由实际网络环境决定。建议最大文件数设为 8。
byLocalOrUTC	BYTE	时间类型: 0-设备本地时间, 即设备 OSD 时间; 1-UTC 时间。
byRes	BYTE[]	保留, 设为 0。最大长度为 63 字节。

A.1.100 NET_EHOME_PIXEL_ARRAY_SIZE

热度图大小信息结构体

成员	数据类型	描述
dwLineValue	DWORD	像素点行值。
dwColumnValue	DWORD	像素点列值。

A.1.101 NET_EHOME_PLANCTRLPARAM

定时控制计划参数结构体

成员	数据类型	描述
enumPlanType	<u>NET_EHOME_PLANCTR</u> <u>L_TYPE</u>	控制类型。
bEnable	BOOL	是否启用控制计划。

参见

NET_EHOME_TERMINAL_CONTROL_V20

A.1.102 NET_EHOME_PLAYBACK_DATA_CB_INFO

回放数据的回调信息结构体

成员	数据类型	描述
dwType	BYTE	数据类型：0-码流头部，1-码流数据。
pData	void*	保存码流头部或码流数据所需的缓冲区。
dwDataLen	DWORD	码流头部或码流数据缓冲区大小。
byRes	BYTE[]	保留。最大长度为 128 字节。

A.1.103 NET_EHOME_PLAYBACK_DATA_CB_PARAM

回放数据的回调参数结构体

成员	数据类型	描述
fnPlayBackDataCB	<u>PLAYBACK_DATA_CB</u>	回放数据的回调函数。
pUserData	void*	用户参数。
byStreamFormat	BYTE	封装格式：0-PS。
byRes	BYTE	保留，设为0。

A.1.104 NET_EHOME_PLAYBACK_INFO_IN

回放请求的输入参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwChannel	DWORD	回放通道号。
byPlayBackMode	BYTE	回放和下载模式：0-按文件名，1-按时间。
byStreamPackage	BYTE	码流回放类型：0-PS（默认），1-RTP。
byLinkMode	BYTE	协议类型：0-TCP，1-UDP（UDP保留），8-NPQ。
byRes	BYTE	保留。
unionPlayBackMode	Union (<u>表16-9</u>)	回放下载模式参数联合体。
struStreamSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	流媒体服务器的地址信息。

表 A-9 回放模式联合体 (unionPlayBackMode)

成员	数据类型	描述
byLen	BYTE	联合体大小, 最大长度为 512 字节。
struPlayBackbyName	Struct (表 16-10)	按文件名回放的参数结构体。
struPlayBackbyTime	Struct (表 16-11)	按时间回放的参数结构体。

表 A-10 按文件名回放的参数结构体

成员	数据类型	描述
szFileName	char	进行回放的文件名。最大长度为 100 字节 (对应宏定义为 “MAX_FILE_NAME_LEN”)。
dwSeekType	DWORD	偏移量计算方式: 0-按字节长度, 1-按时间 (秒数)。
dwFileOffset	DWORD	文件偏移量, 开始回放或下载的位置。
dwFileSpan	DWORD	下载的文件大小, 为 0 时, 表示下载整个文件。

表 A-11 按时间回放的参数结构体

成员	数据类型	描述
struStartTime	<i>NET_EHOME_TIME</i>	开始回放的时间。
struStopTime	<i>NET_EHOME_TIME</i>	停止回放的时间。
byLocalOrUTC	BYTE	时间类型: 0-设备本地时间, 即设备 OSD 时间; 1-UTC 时间。
byDuplicateSegment	BYTE	重复时间段的位置: 0-重复时间段的前段, 1-重复时间段的

成员	数据类型	描述
		后段。当 byLocalOrUTC 为 1 时，该参数无效。

备注

目前仅支持按文件名回放。

A.1.105 NET_EHOME_PLAYBACK_INFO_OUT

回放请求的输出参数结构体

成员	数据类型	描述
iSessionID	LONG	设备返回的回放请求会话 ID：0-无效。
lHandle	LONG	异步回调的消息句柄。
byRes	BYTE[]	保留，设为 0。最大长度为 124 字节。

A.1.106 NET_EHOME_PLAYBACK_LISTEN_PARAM

回放的监听参数结构体

成员	数据类型	描述
strulPAdress	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	本地监听信息。IP 为 0.0.0.0 的情况下，默认为本地地址；当存在多个网卡时，默认采用从操作系统获取到的第一个地址。
fnNewLinkCB	<u>PLAYBACK_NEWLINK_C</u> <u>B</u>	回放请求的回调函数。
pUserData	void*	用户参数。

成员	数据类型	描述
byLinkMode	BYTE	监听请求的接入协议类型：0-TCP，1-UDP（保留），8-NPQ。
byRes	BYTE[]	保留，设为0。最大长度为127字节。

备注

若设备不支持 4.0 及以上版本的 ISUP，则 **ISessionID** 无效（设为 0）。请使用不同的端口和回调函数开启监听，以区别所接入的设备和通道。

A.1.107 NET_EHOME_PLAYBACK_NEWLINK_CB_INFO

回放请求的回调信息结构体（适用于 64 位 Windows 或 Linux 操作系统）

成员	数据类型	描述
szDeviceID	char[]	设备 ID。最大长度为 256 字节（对应宏定义为“NET_EHOME_DEVICEID_LEN”）。
ISessionID	LONG	回放请求的设备会话 ID，由 <i>NET_ECMS_StartPlayBack</i> 返回。若无效，则设为 0。
dwChannelNo	DWORD	设备通道号，0-无效。
sDeviceSerial	char[]	设备序列号。最大长度为 12 字节（对应宏定义为“NET_EHOME_SERIAL_LEN”）。
byStreamFormat	BYTE	码流封装格式：0-PS，1-RTP。
byRes1	BYTE[]	保留，最大长度为 3 字节。
fnPlayBackDataCB	<i>PLAYBACK_DATA_CB</i>	数据回调函数，其大小为 8 字节。
pUserData	void*	用户参数，其大小为 8 字节。
byRes	BYTE[]	保留，设为 0。最大长度为 88 字节。

回放请求的回调信息结构体（适用于 32 位 Windows 或 Linux 操作系统）

成员	数据类型	描述
szDeviceID	char[]	设备 ID。最大长度为 256 字节（对应宏定义为“NET_EHOME_DEVICEID_LEN”）。
lSessionID	LONG	回放请求的设备会话 ID，由 <i>NET_ECMS StartPlayBack</i> 返回。若无效，则设为 0。
dwChannelNo	DWORD	设备通道号，0-无效。
sDeviceSerial	char[]	设备序列号。最大长度为 12 字节（对应宏定义为“NET_EHOME_SERIAL_LEN”）。
byStreamFormat	BYTE	码流封装格式：0-PS，1-RTP。
byRes1	BYTE[]	保留，最大长度为 3 字节。
fnPlayBackDataCB	<i>PLAYBACK_DATA_CB</i>	数据回调函数，其大小为 4 字节。
byRes2	BYTE[]	保留，最大长度为 4 字节。
pUserData	void*	用户参数，其大小为 4 字节。
byRes3	BYTE[]	保留，最大长度为 4 字节。
byRes	BYTE[]	保留，设为 0。最大长度为 88 字节。

A.1.108 NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM

回放暂停或恢复参数结构体

成员	数据类型	描述
ISessionID	LONG	回放的会话 ID。
IHandle	LONG	输出参数指针, 在异步模式中作为异步回调的标识, 对应 <u>NET_EHOME_CMSCB_DATA</u> 中的 dwHandle 。
byRes	BYTE[]	保留。最大长度为 120 字节。

A.1.109 NET_EHOME_PLAYER_PARAM

终端播放参数结构体

成员	数据类型	描述
szCfgType	BYTE[]	配置类型：“volume”（音量），“light”（亮度），“logo”（LOGO），“defaultSchedule”（默认计划），“password”（密码），“temperature”（温度），“allParam”（所有参数）。最大长度为 32 字节。
dwplayerVolume	DWORD	播放音量。
struDisplayParam	<u>NET_EHOME_DISPLAY_PARAM</u>	显示参数。
struPasswordCfg	<u>NET_EHOME_PASSWORD_CFG</u>	终端密码。

A.1.110 NET_EHOME_POINT

坐标参数结构体

成员	数据类型	描述
dwX	DWORD	X 轴坐标信息。
dwY	DWORD	Y 轴坐标信息。
byRes	BYTE[]	保留，设为 0。最大长度为 4 字节。

A.1.111 NET_EHOME_POSITION_INFO

位置信息结构体

成员	数据类型	描述
dwPositionX	DWORD	位置的 X 轴坐标。
dwPositionY	DWORD	位置的 Y 轴坐标。
dwHeight	DWORD	高度。
dwWidth	DWORD	宽度。

A.1.112 NET_EHOME_POST_PUBLISH_XML

XML 文件传输参数结构体，用于配置配置日程、插播、升级等参数

成员	数据类型	描述
szServerIP	BYTE[]	服务器 IP 地址。最大长度为 32 字节。
szServerPort	DWORD	服务器端口号。
dwTerminalID	DWORD	终端 ID，随进程返回。
dwXmlUniqueSeq	DWORD	XML 文件 ID。
dwXmlSizeH	DWORD	XML 文件大小，高 32 位表示。

成员	数据类型	描述
dwXmlSizeL	DWORD	XML 文件大小，低 32 位表示。
byStorageId	BYTE[]	XML 文件的保存路径。最大长度为 128 字节。
byRes	BYTE[]	保留。最大长度为 16 字节。

A.1.113 NET_EHOME_POST_SCHEDULE

日程下发参数结构体

成员	数据类型	描述
byServerIP	BYTE[]	服务器 IP 地址。最大长度为 32 字节。
dwServerPort	DWORD	服务器端口号。
dwScheduleID	DWORD	新建日程是分配的日程 ID。
dwScheduleSeq	DWORD	标示日程是否被修改，是否可以继续传输，只针对 ANR 有效。
byIsDefaultSchedual	BOOL	标示是否为垫片日程。
byRes	BYTE[]	保留。最大长度为 2 字节。
dwTimingSchedualID	DWORD	日程 ID。
byEffectiveTime	BYTE[]	日程开始播放的时间。最大长度为 32 字节。

A.1.114 NET_EHOME_PPPOECFG

PPPoE 参数结构体

成员	数据类型	描述
dwPPPoE	DWORD	是否启用 PPPoE : 1-是, 0-否。
sPPPoEUser	char[]	PPPoE 用户名。最大长度为 32 字节 (对应宏定义为 “NET_EHOME_SERIAL_LEN”)。
sPPPoEPassword	char[]	PPPoE 密码。最大长度为 16 字节 (对应宏定义为 “PASSWD_LEN”)。
struPPPoEIP	<u>NET_EHOME_IPADDR</u>	PPPoE 的 IP 地址。

A.1.115 NET_EHOME_PRESET_PARAM**预置点参数结构体**

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byPresetCmd	BYTE	预置点控制命令 : 1-设置预置点, 2-删除预置点, 3-调用预置点。
byRes1	BYTE[]	保留, 设为 0。最大长度为 3 字节。
dwPresetIndex	DWORD	预置点编号。
byRes2	BYTE[]	保留, 设为 0。最大长度为 32 字节。

A.1.116 NET_EHOME_PREVIEW_CB_MSG

预览回调数据结构体

成员	数据类型	描述
byDataType	BYTE	数据类型：1-NET_DVR_SYSHEAD（码流头部），2-NET_DVR_STREAMDATA（码流数据）。
byRes1	BYTE[]	保留。最大长度为 3 字节。
pRecvdata	void*	保存码流头部或码流数据的缓冲区。
dwDataLen	DWORD	码流头部或码流数据缓冲区大小。
byRes2	BYTE[]	保留。最大长度为 128 字节。

A.1.117 NET_EHOME_PREVIEW_DATA_CB_PARAM


预览回调参数结构体

成员	数据类型	描述
fnPreviewDataCB	<u>PREVIEW_DATA_CB</u>	预览的回调函数。
pUserData	void*	用户参数。
byStreamFormat	BYTE	码流封装格式：0-PS
byRes	BYTE[]	保留，设为 0。最大长度为 127 字节。

A.1.118 NET_EHOME_PREVIEWINFO_IN

预览请求的输入参数结构体

成员	数据类型	描述
iChannel	int	通道号。
dwStreamType	DWORD	码流类型：0-主码流，1-子码流，2-第三码流。

成员	数据类型	描述
dwLinkMode	DWORD	接入方式：0-TCP，1-UDP，2-HRUDP（可靠传输），8-NPQ  说明 NPQ 功能的使用，目前和 Smart、变码率的配置是互斥的，如果要使用 NPQ 功能，必须将码率调整为定码率，关闭 Smart 功能，8111 错误码作为修改配置的错误提示。
struStreamSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	流媒体服务器的地址信息。

A.1.119 NET_EHOME_PREVIEWINFO_IN_V11

预览请求的输入参数结构体

成员	数据类型	描述
iChannel	int	通道号。
dwStreamType	DWORD	码流类型：0-主码流，1-子码流，2-第三码流。
dwLinkMode	DWORD	接入方式：0-TCP，1-UDP，2-HRUDP（可靠传输）
struStreamSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	流媒体服务器的地址信息。
byDelayPreview	BYTE	预览模式：0-获取实时码流，1-获取延迟码流。
byEncrypt	BYTE	码流是否加密：0-否，1-是。
byRes	BYTE[]	保留，设为 0。最大长度为 30 字节。

A.1.120 NET_EHOME_PREVIEWINFO_OUT

预览请求的输出参数结构体

成员	数据类型	描述
lSessionID	int	预览请求的会话 ID。
lHandle	DWORD	异步取流回调的消息句柄。
byRes	BYTE[]	保留，设为 0。最大长度为 124 字节。

A.1.121 NET_EHOME_PT_PARAM

监听代理参数结构体

成员	数据类型	描述
strulP	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	监听地址信息（IP 地址和端口号）。
byProtocolType	BYTE	协议类型：0-TCP
byProxyType	BYTE	代理类型：0-海康私有协议代理，1-HTTP 代理。
byRes	BYTE[]	保留。最大长度为 2 字节。

A.1.122 NET_EHOME_PTXML_PARAM

传输参数结构体

成员	数据类型	描述
pRequestUrl	void*	请求 URL。
dwRequestUrlLen	DWORD	请求 URL 大小。
pCondBuffer	void*	保存条件参数（XML 或 JSON 格式）的缓冲区。

成员	数据类型	描述
dwCondSize	DWORD	条件缓冲区大小。
pInBuffer	void*	保存输入参数（XML 或 JSON 格式）的缓冲区。
dwInSize	DWORD	输入缓冲区大小。
pOutBuffer	void*	保存输出参数（XML 或 JSON 格式）的缓冲区。
dwOutSize	DWORD	输出缓冲区大小。输出缓冲区开辟的空间充足时，表示返回的报文数据大小。
dwReturnedXMLLen	DWORD	实际接收 XML 或 JSON 数据大小。输出缓冲区开辟的空间不足时，表示返回的报文数据大小。
dwRecvTimeOut	DWORD	接收超时时间。
dwHandle	DWORD	消息句柄。设置了回放异步回调之后，在回调中用于标识。
byRes	BYTE[]	保留，设为 0。最大长度为 24 字节。

A.1.123 NET_EHOME_PTZ_PARAM

PTZ 参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byPTZCmd	BYTE	PTZ 控制命令，参见备注。
byAction	BYTE	PTZ 控制：0-开始，1-停止。

成员	数据类型	描述
bySpeed	BYTE	PTZ 速度，取值范围从 0 到 70。值越大，代表速度越快。
byRes	BYTE[]	保留。最大长度为 29 字节。

PTZ 控制命令

dwCommand	命令号	描述
PTZ_UP	0	向上
PTZ_DOWN	1	向下
PTZ_LEFT	2	向左
PTZ_RIGHT	3	向右
PTZ_UPLEFT	4	向左上
PTZ_DOWNLEFT	5	向左下
PTZ_UPRIGHT	6	向右上
PTZ_DOWNRIGHT	7	向右下
PTZ_ZOOMIN	8	变焦（缩小）
PTZ_ZOOMOUT	9	变焦（放大）
PTZ_FOCUSNEAR	10	聚焦-
PTZ_FOCUSFAR	11	聚焦+
PTZ_IRISSTARTUP	12	光圈变大
PTZ_IRISSTOPDOWN	13	光圈变小
PTZ_LIGHT	14	补光灯
PTZ_WIPER	15	雨刷
PTZ_AUTO	16	自动

A.1.124 NET_EHOME_PUBLISH_SERVERADDR

信息发布服务器的地址信息结构体

成员	数据类型	描述
enumAddressingForm atType	<i>NET_EHOME_ADDRFO</i> <i>RMAT_CMD</i>	地址类型。
szIpAddress	BYTE[]	IP 地址。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
szIpv6Address	BYTE[]	IPv6 地址。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
szHostName	BYTE[]	域名。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
dwPortNo	DWORD	端口号。
szUserName	BYTE[]	用户名。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
szPassword	BYTE[]	密码。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
bRegStatus	BOOL	注册状态。

A.1.125 NET_EHOME_PUSHPLAYBACK_IN

回放码流传输请求的输入参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lSessionID	LONG	会话 ID, 由 <i>NET_ECMS_StartPlayBack</i> 返回。

成员	数据类型	描述
byKeyMD5	BYTE[]	码流加密密钥，通过两次 MD5 计算获得。最大长度为 32 字节。
byRes	BYTE[]	保留，设为 0。最大长度为 96 字节。

A.1.126 NET_EHOME_PUSHPLAYBACK_OUT

回放码流传输请求的输出参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lHandle	LONG	异步回调的消息句柄。
byRes	BYTE[]	保留，设为 0。最大长度为 124 字节。

A.1.127 NET_EHOME_PUSHSTREAM_IN

实时流传输请求的输入参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lSessionID	LONG	预览请求的会话 ID，由 <i>NET_ECMS_StartGetRealStreamV11</i> 返回。
byRes	BYTE[]	保留，设为 0。最大长度为 128 字节。

A.1.128 NET_EHOME_PUSHSTREAM_OUT

实时流传输请求的输出参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lHandle	LONG	异步取流回调的消息句柄。
byRes	BYTE[]	保留，设为 0。最大长度为 124 字节。

A.1.129 NET_EHOME_PUSHVOICE_IN

音频码流传输请求的输入参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lSessionID	LONG	语音对讲请求的会话 ID，由 <i>NET_ECMS_StartVoiceWithStmServer</i> 返回。
byToken	BYTE[]	令牌信息。最大长度为 64 字节。
byRes	BYTE[]	保留，设为 0。最大长度为 64 字节。

A.1.130 NET_EHOME_PUSHVOICE_OUT

音频码流传输请求的输出参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lHandle	LONG	异步音频码流回调的消息句柄。
byRes	BYTE[]	保留，设为 0。最大长度为 124 字节。

A.1.131 NET_EHOME_PZIN_PARAM

变倍参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
byAction	BYTE	变倍类型：0-缩小，1-放大。
byRes1	BYTE[]	保留，设为0。最大长度为3字节。
struArea	<u>NET_EHOME_ZONE</u>	框选范围参数。
byRes2	BYTE[]	保留，设为0。最大长度为32字节。

A.1.132 NET_EHOME_QUEUE_DATA

呼叫号码数据结构体

成员	数据类型	描述
szQueueDataValue	char[]	呼叫号码。最大长度为32字节。
dwQueueDataId	DWORD	呼叫号码ID。

A.1.133 NET_EHOME_QUEUE_DATA_LIST

呼叫号码列表结构体

成员	数据类型	描述
queueData	<u>NET_EHOME_QUEUE_DATA</u> []	呼叫号码。最大长度为20字节。

A.1.134 NET_EHOME_QUEUE_INFO

叫号管理信息结构体

成员	数据类型	描述
dwMaterialId	DWORD	素材 ID。
dwQueueId	DWORD	呼叫号码 ID。
enumDataType	<u>NET_EHOME_THIRD_P</u> <u>ARTY_DATA_TYPE</u>	命令类型。
enumRefreshType	<u>NET_EHOME_REFRESH</u> <u>_TYPE</u>	数据刷新类型。
struItemDataList	<u>NET_EHOME_QUEUE_I</u> <u>TEM_DATALIST</u>	呼叫号码列表。

A.1.135 NET_EHOME_QUEUE_ITEM_DATALIST

呼叫号码列表结构体

成员	数据类型	描述
dwQueueDataListCnt	DWORD	列表的行数。
dwQueueDataCnt	DWORD	列表的列数。
struQueueDataList	<u>NET_EHOME_QUEUE</u> <u>DATALIST []</u>	呼叫号码列表。最大长度为 4 字节。

A.1.136 NET_EHOME_REC_FILE

视频文件查找结果结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
szFileName	char[]	视频文件名。最大长度为 100 字节(对应宏定义为“MAX_FILE_NAME_LEN”)。
struStartTime	<u>NET_EHOME_TIME</u>	开始时间。
struStopTime	<u>NET_EHOME_TIME</u>	结束时间。
dwFileSize	DWORD	视频文件大小。
dwFileMainType	DWORD	录像类型：0xff-全部录像类型，0-持续录像，1-移动侦测录像，2-报警触发录像，3-报警触发或移动侦测录像，4-报警和移动侦测录像，5-命令触发录像，6-手动录像，7-震动报警录像，8-环境报警录像，9-智能报警录像（或取证录像），10（0x0a）-PIR 报警录像，11（0x0b）-无线报警录像，12（0x0c）-呼救报警录像，13（0x0d）-所有种类报警录像。
dwFileSubType	DWORD	视频文件的子类型：0-非交通取证视频，1-超车，2-占道，3-违章停车，4-逆行，5-倒车，6-掉头，7-变道，8-压线，9-其他；只有当 dwFileMainType 设为 1 时该参数有效。
dwFileIndex	DWORD	视频文件编号，从 0 开始。
byTimeDiffH	BYTE	开始/结束时间与 UTC 时间差（小时），取值为-12，-10，…，+12，+14，0xff 表示无效。
byTimeDiffM	BYTE	开始/结束时间与 UTC 时间差（分钟），取值为-30，0，30，45，0xff 表示无效。
byRes	BYTE[]	保留，设为 0。最大长度为 126 字节。

A.1.137 NET_EHOME_REC_FILE_COND

视频文件查找条件结构体

成员	数据类型	描述
dwChannel	DWORD	通道号。
dwRecType	DWORD	录像类型：0xff-全部录像类型，0-持续录像，1-移动侦测录像，2-报警触发录像，3-报警触发或移动侦测录像，4-报警和移动侦测录像，5-命令触发录像，6-手动录像，7-震动报警录像，8-环境报警录像，9-智能报警录像（或取证录像），10（0x0a）-PIR 报警录像，11（0x0b）-无线报警录像，12（0x0c）-呼救报警录像，13（0x0d）-所有种类报警录像。
struStartTime	<i>NET_EHOME_TIME</i>	开始时间。
struStopTime	<i>NET_EHOME_TIME</i>	结束时间。
dwStartIndex	DWORD	查询的起始位置，从 0 开始。
dwMaxFileCountPer	DWORD	单次搜索可查询的最大文件数，由实际网络环境决定。建议最大文件数设为 8。
byLocalOrUTC	BYTE	时间类型：0-设备本地时间，即设备 OSD 时间；1-UTC 时间。
byRes	BYTE[]	保留，设为 0。最大长度为 63 字节。

A.1.138 NET_EHOME_RECORD_CHAN

关联的录像通道参数结构体

成员	数据类型	描述
byAnalogChanNum	BYTE	模拟通道数量，只读。
byAnalogChan	BYTE[]	模拟通道状态，用数组下标表示：0-未启用，1-启用。最大长度为 32 字节（对应宏定义为“MAX_ANALOG_CHANNUM”）。
byRes1	BYTE[]	保留，设为 0。最大长度为 3 字节。
wDigitChanNum	WORD	数字通道数量，只读。
byDigitChan	BYTE[]	数字通道状态，用数组下标表示，0-未启用，1-启用。最大长度为 480 字节。
byRes2	BYTE[]	保留，设为 0。最大长度为 62 字节。

A.1.139 NET_EHOME_REGISTER_LISTEN_MODE

监听模式结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwRegisterListenMode	DWORD	监听模式：0-TCP 和 UDP、1-UDP、2-TCP。
byRes	BYTE[]	保留。最大取值长度为 128 字节。

A.1.140 NET_EHOME_RELATE_IPC_INFO

关联的网络摄像机信息结构体

成员	数据类型	描述
dwIpcNum	DWORD	关联的网络摄像机数量。
strulpcInfo	<u>NET_EHOME_IPC_INF</u> 0 []	网络摄像机信息。最大长度为 6 字节。

A.1.141 NET_EHOME_RELEASE_SERVER_INFO

信息发布系统信息结构体

成员	数据类型	描述
dwKeepAliveSeconds	DWORD	保活时间，单位：秒。
byAlarmServerIP	BYTE[]	报警管理服务器的 IP 地址。最大长度为 32 字节。
dwAlarmServerPort	DWORD	报警管理服务器的端口号。
dwAlarmServerType	DWORD	报警管理服务器的通信类型：0-TCP（默认），1-UDP。
dwAlarmServerTcpPort	DWORD	报警管理服务器下发给终端下发的报警端口号。
byNtpServerIP	BYTE[]	NTP 服务器的 IP 地址。最大长度为 32 字节。
dwNtpServerPort	DWORD	NTP 服务器的端口号。
dwNtpInterval	DWORD	NTP 时间同步的间隔。
byPicServerIP	BYTE[]	存储服务器的 IP 地址。最大长度为 32 字节。
dwPicServerPort	DWORD	存储服务器的端口号。
dwPicServerType	DWORD	存储服务器的通信类型：0-TCP（默认），1-UDP。

成员	数据类型	描述
byBlackListAddr	BYTE[]	黑名单管理服务器的地址。最大长度为 32 字节。
byBlackListName	BYTE[]	黑名单管理服务器的名称。最大长度为 32 字节。
dwBlackListPort	DWORD	黑名单管理服务器的端口号。
byBlackListUser	BYTE[]	黑名单管理服务器的用户名。最大长度为 32 字节。
byBlackListPasswd	BYTE[]	黑名单管理服务器的密码。最大长度为 32 字节。
dwTranserialSvrPort	DWORD	传输服务器的端口号。
bReliableTransmission	BOOL	是否开启可靠传输：True-是， False-否。

A.1.142 NET_EHOME_REMOTE_CTRL_PARAM

控制参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lpCondBuffer	void*	保存条件参数的缓冲区，由 <u>NET_ECMS RemoteControl</u> 的控制命令 (dwCommand) 决定，请参见备注中的表格。
dwCondBufferSize	DWORD	条件参数缓冲区大小。
lpInbuffer	void*	保存控制参数的缓冲区，由 <u>NET_ECMS RemoteControl</u> 的 (dwCommand) 决定，请参见备注中的表格。

成员	数据类型	描述
dwInBufferSize	DWORD	控制参数缓冲区大小。
byRes	BYTE[]	保留，设为 0。最大长度为 32 字节。

备注

dwCommand、**IpCondBuffer** 和 **IpInbuffer** 之间的关系请参见下表。

dwCommand	IpCondBuffer	IpInbuffer
NET_EHOME_MANUAL_IOOUT	Null	<u>NET_EHOME_MANUAL_IOOUT_CTRL</u>
NET_EHOME_PTZ_CTRL	4 字节通道号 (long)	<u>NET_EHOME_PTZ_PARAM</u>
NET_EHOME_PRESET_CTRL	4 字节通道号 (long)	<u>NET_EHOME_PRESET_PARAM</u>
NET_EHOME_PZIN	4 字节通道号 (long)	<u>NET_EHOME_PZIN_PARAM</u>
NET_EHOME_PTRACK	4 字节通道号 (long)	<u>NET_EHOME_POINT</u>

A.1.143 NET_EHOME_REPLACE_MATERIAL

素材更换参数结构体

成员	数据类型	描述
szServerIP	BYTE[]	服务器 IP 地址。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
dwServerPort	DWORD	服务器端口号。
dwMaterialId	DWORD	素材 ID。
dwMaterialSeq	DWORD	素材编号。

A.1.144 NET_EHOME_SCHEDTIME

时间段参数结构体

成员	数据类型	描述
byStartHour	BYTE	开始时间：时。
byStartMin	BYTE	开始时间：分。
byStopHour	BYTE	结束时间：时。
byStopMin	BYTE	结束时间：分。

A.1.145 NET_EHOME_SCREEN_SHOT

终端截屏参数结构体

成员	数据类型	描述
dwServerPort	DWORD	服务器端口号。
dwClientFdIndex	DWORD	客户端 FD 索引号。

A.1.146 NET_EHOME_SCREEN_SHOT_EX

终端截屏参数结构体

成员	数据类型	描述
dwServerPort	DWORD	服务器端口号。
dwClientFdIndex	DWORD	客户端 FD 索引号。
szKmsURL	BYTE[]	截屏图片在 KMS 服务器上的 URL 地址。最大长度为 256 字节。

A.1.147 NET_EHOME_SCREEN_SHOT_RET

终端截屏响应结构体

成员	数据类型	描述
dwConnfd	unsigned int	终端和服务器之间异步通信的套接字。
dwSeq	unsigned int	终端和服务器之间异步通信的 ID。
szUUID	char[]	截屏图片的 UUID。最大长度为 64 字节。
szPicURL	char[]	截屏图片在 KMS 服务器中的 URL 地址。最大长度为 256 字节。

A.1.148 NET_EHOME_SEND_PARAM

数据发送参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwRecvTimeOut	DWORD	接收超时时间，单位：ms。取值范围为 0 或 [5000,60000]。当为 0 时，表示使用默认接收超时时间（5000 ms）。
bySendTimes	BYTE	报文重复发送次数，在网络较差时可减少丢包的情况。取值范围为 1、2、3。默认为 1 次，最大 3 次。
byRes2	BYTE[]	保留。最大长度为 127 字节。

A.1.149 NET_EHOME_SERVER_INFO

服务器信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwKeepAliveSec	DWORD	心跳间隔，单位：秒，默认：15 秒。
dwTimeOutCount	DWORD	心跳超时次数，该值默认为 0 时，表示心跳超时次数为 6 次。
struTCPAlarmSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	报警管理服务器的 TCP 地址。
struUDPAlarmSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	报警管理服务器的 UDP 地址。
dwAlarmServerType	DWORD	报警管理服务器（AMS）类型：0-只支持 UDP，1-支持 UDP 和 TCP，2-支持 MQTT（服务器地址基于 TCP）。
struNTPSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	NTP 服务器地址信息。
dwNTPInterval	DWORD	NTP 时间同步的间隔，单位：秒。
struPictureSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	存储服务器的地址信息。
dwPicServerType	DWORD	存储服务器（SS）类型：0-Tomcat，1-VRB，2-云存储，3-KMS，4-ISUP5.0。
struBlackListServer	<u>NET_EHOME_BLACKLIS</u> <u>T_SEVER</u>	黑名单管理服务器。
byRes	BYTE[]	保留，设为 0。最大长度为 128 字节。

A.1.150 NET_EHOME_SERVER_INFO_V50

服务器信息结构体（V50）

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwKeepAliveSec	DWORD	心跳间隔，单位：秒，默认：15s。
dwTimeOutCount	DWORD	心跳超时次数，该值默认为 0 时，表示心跳超时次数为 6 次。
struTCPAlarmSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	报警管理服务器的 TCP 地址。
struUDPAlarmSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	报警管理服务器的 UDP 地址。
dwAlarmServerType	DWORD	报警管理服务器（AMS）类型：0-只支持 UDP，1-支持 UDP 和 TCP，2-支持 MQTT（服务器地址基于 TCP）。
struNTPSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	NTP 服务器地址信息。
dwNTPInterval	DWORD	NTP 时间同步的间隔，单位：秒。
struPictureSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	存储服务器的地址信息。
dwPicServerType	DWORD	存储服务器（SS）类型：0-Tomcat，1-VRB，2-云存储，3-KMS，4-ISUP5.0。
struBlackListServer	<u>NET_EHOME_BLACKLIS</u> <u>T_SEVER</u>	黑名单管理服务器的参数。
struRedirectSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	重定向服务器的地址。
byClouldAccessKey	BYTE[]	云存储的访问密码。最大长度为 64 字节。
byClouldSecretKey	BYTE[]	云存储的密钥。最大长度为 64 字节。

成员	数据类型	描述
byCloudHttps	BYTE	启用云存储类型：1-HTTPS，0-HTTP。
byRes1	BYTE[]	保留。最大长度为 3 字节。
dwAlarmKeepAliveSec	DWORD	报警心跳间隔，单位：秒，默认：30s。
dwAlarmTimeOutCount	DWORD	报警心跳超时次数，默认设为 0，表示报警心跳超时次数为 3 次。
dwCloudPoolId	DWORD	云存储池 ID。
byRes	BYTE[]	保留。最大长度为 368 字节。

A.1.151 NET_EHOME_SET_REREGISTER_MODE

设备注册模式结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwReRegisterMode	DWORD	是否支持重复注册（仅支持 2.0 版本 ISUP）： 0-在保活时间内允许 2.0 协议设备重复进行注册（默认），1-在保活时间内不允许 2.0 协议设备重复进行注册，返回错误码“403”。

A.1.152 NET_EHOME_SHOT_PIC

弹图命令参数结构体

成员	数据类型	描述
byServerIP	BYTE[]	服务器 IP 地址。最大长度为 32 字节。
dwServerPort	DWORD	服务器端口号。
dwShotPicID	DWORD	弹图 ID。

A.1.153 NET_EHOME_SS_CENTRAL_PARAM

中心存储联合体

成员	数据类型	描述
pPoolId	const char*	资源池 ID，最大长度为 32 字节。
pSerialID	const char*	设备 ID，最大长度为 64 字节。
pSerialIDUUID	const char*	SerialID+UUID 表示视频文件的 key，最大长度为 64 字节（设备 ID 长度）+32 字节（UUID 长度）。
pBeginTime	const char*	开始时间。
pEndTime	const char*	结束时间。
byPoolIdLength	BYTE	资源池 ID 长度，仅上传文件时生效。
bySerialIDLength	BYTE	设备 ID 长度，仅上传文件时生效。
bySerialIDUUIDLength	BYTE	视频文件 key 长度，仅文件类型为视频文件时有效。
byBeginTimeLength	BYTE	开始时间长度。
byEndTimeLength	BYTE	结束时间长度。
byTransform	BYTE	是否需要转封装：0-不需要（默认）、1-需要。
byRes1	BYTE[]	保留，最大长度为 2 字节。

成员	数据类型	描述
dwRecordType	int	录像类型。
dwSourceDataType	int	视频文件类型：4-海康视频文件（存储内部进行帧分析）、其它值-第三方视频文件。
dwHeadSize	int	录像头长度。
dwErrorCode	int	中心存储错误码。
byRes	BYTE[]	保留。最大长度为 468 字节。

A.1.154 NET_EHOME_SS_CLIENT_PARAM

文件上传客户端参数结构体

成员	数据类型	描述
enumType	<i>NET_EHOME_SS_CLIENT_TYPE</i>	文件上传客户端类型。
struAddress	<i>NET_EHOME_IPADDRESS</i>	存储服务器（SS）的 IP 地址。
byHttps	BYTE	是否启用 HTTPS 通信。
byRes	BYTE[]	保留，设为 0。最大长度为 63 字节。

相关接口

NET_ESS_CreateClient

A.1.155 NET_EHOME_SS_CLOUD_PARAM

云存储结构体

成员	数据类型	描述
pPoolId	const char*	资源池 ID。
byPoolIdLength	BYTE	资源池 ID 长度。
dwErrorCode	int	云存储错误码。
byRes	BYTE[]	保留，最大长度为 503 字节。

A.1.156 NET_EHOME_SS_EX_PARAM

回调函数返回的扩展结构体

成员	数据类型	描述
byProtoType	BYTE	存储协议类型 1-云存储，2-TOMCAT，3-KMS，4-VRB。
byRes	BYTE[]	保留，最大长度为 23 字节。
<u>表 16-12</u>	union	联合体。

表 A-12 unionStoreInfo

成员	描述
struCentral	<u>NET_EHOME_SS_CENTRAL_PARAM</u>
struCloud	<u>NET_EHOME_SS_CLOUD_PARAM</u>
struTomcat	<u>NET_EHOME_SS_TOMCAT_PARAM</u>
struKms	<u>NET_EHOME_SS_KMS_PARAM</u>
struVrb	<u>NET_EHOME_SS_VRB_PARAM</u>

A.1.157 NET_EHOME_SS_INIT_PARAM

初始化参数结构体

成员	数据类型	描述
szKey	char[]	数据库密钥（字符串类型），最大长度为 32 字节（对应宏定义为“SS_DB_KEY_MAX_LEN”）。
byRes	BYTE[]	保留，最大长度为 224 字节。

备注

- 允许 NET_ESS_Init_V11 接口调用时传递 NULL 参数，此时等同于调用 NET_ESS_Init 接口。即不设置数据库密钥。
- HCISUPSS 生成的数据库为 PictrueStorageServer.db。当运行时已存在该数据库，若该数据库进行了加密（已设置了密钥），则必须调用 NET_ESS_Init_V11 接口，并传递正确的密钥，才能完成 HCISUPSS 的初始化。
- 当运行时若已存在该数据库，若该数据库未进行加密（未设置密钥），不可以通过调用 NET_ESS_Init_V11 对已有的非加密数据库进行密钥设置，只能通过非加密的方式打开。
- 允许存储组件在初始化失败之后调用 NET_ESS_GetLastError 来获取错误码，如调用 NET_ESS_Init_V11 未设置正确的密码，则调用 NET_ESS_GetLastError 获取到的错误码为 1（NET_DVR_PASSWORD_ERROR）（密码错误）。当数据库已经设置了密钥，但调用 NET_ESS_Init 传递空的密钥，因此错误码还是为 1（NET_DVR_PASSWORD_ERROR）（密码错误）。
- 不允许 **szKey** 为空，否则 NET_ESS_Init_V11 调用失败，错误码为参数错误（错误码 17）。

A.1.158 NET_EHOME_SS_KMS_PARAM

KMS 存储信息结构体

成员	数据类型	描述
byRes	BYTE[]	保留，最大长度为 512 字节。

A.1.159 NET_EHOME_SS_LISTEN_HTTPS_PARAM

HTTPS 监听参数结构体

成员	数据类型	描述
byHttps	BYTE	是否启用 HTTPS: 0-不启用, 1-启用。
byVerifyMode	BYTE	校验模式: 0-单项校验。
byCertificateFileType	BYTE	证书类型: 0-PEM, 1-ANS1
byPrivateKeyFileType	BYTE	私钥类型: 0-PEM, 1-ANS1
szUserCertificateFile	char[]	公钥证书存储路径。最大长度为 260 字节(对应宏定义为“MAX_PATH”)。
szUserPrivateKeyFile	char[]	私钥证书存储路径。最大长度为 260 字节(对应宏定义为“MAX_PATH”)。
dwSSLVersion	DWORD	SSL 模式: 0-SSL23, 1-SSL2, 2-SSL3, 3-TLS1.0, 4-TLS1.1, 5-TLS1.2。SSL2 和 SSL3 模式兼容, 可与最新版本的客户端和服务端匹配。
byRes3	BYTE[]	保留。最大长度为 360 字节。

A.1.160 NET_EHOME_SS_LISTEN_PARAM

监听参数结构体

成员	数据类型	描述
struAddress	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	本地监听信息。IP 为 0.0.0.0 的情况下, 默认为本地地址; 当存在多个网卡时, 默认采用从操作系统获取到的第一个地址。
szKMS_UserName	char[]	KMS 的 username。最大长度为 512 字节(对应宏定义为“MAX_KMS_USER_LEN”)。
szKMS_Password	char[]	KMS 的 password。最大长度为 512 字节(对应宏定义为“MAX_KMS_USER_LEN”)。

成员	数据类型	描述
fnSStorageCb	<u><i>EHomeSSMsgCallBack</i></u>	存储信息的回调函数。
fnSMsgCb	<u><i>EHomeSSMsgCallBack</i></u>	存储文件的回调函数。
szAccessKey	char[]	ISUP5.0 的 AccessKey。最大长度为 64 字节（对应宏定义为“MAX_CLOUD_AK_SK_LEN”）。
szSecretKey	char[]	ISUP5.0 的 SecretKey。最大长度为 64 字节（对应宏定义为“”）。
pUserData	void*	用户参数。
byHttps	BYTE	是否启用 HTTPS 通信：0-不启用，1-启用。
byRes1	BYTE[]	保留。最大长度为 3 字节。
fnSSRWCb	<u><i>EHomeSSRWCallBack</i></u>	读写的回调函数。
fnSSRWCbEx	<u><i>EHomeSSRWCallBackE</i></u> x	读写回调函数扩展。
bySecurityMode	BYTE	安全模式开关，默认开启，0-开启，1-关闭。
byRes	BYTE[]	保留，设为 0。最大长度为 51 字节。

备注

读写回调函数（*EHomeSSRWCallBack*）和存储回调函数（*EHomeSSStorageCallBack*）是互斥的，不可同时使用。如果两个回调函数都进行了注册和配置，那么默认 *EHomeSSRWCallBack* 生效。

相关接口

NET_ECMS_StartListen

A.1.161 NET_EHOME_SS_LOCAL_SDK_PATH

存储服务器的本地加载路径结构体

成员	数据类型	描述
sPath	char[]	加载路径。最大长度为 260 字节(对应宏定义为“MAX_PATH”)。
byRes	BYTE[]	保留。最大长度为 128 字节。

A.1.162 NET_EHOME_SS_RW_PARAM

读写扩展回调函数结构体

成员	数据类型	描述
pFileName	const char*	文件名。
pFileBuf	void*	文件内容。
dwFileLen	DWORD*	文件大小。
pFileUrl	const char*	文件 URL。  说明 KMS 存储方式获取到的 URL 中含有 token 参数，用于下载文件前须删除该 token 参数。
pUser	void*	用户数据。
byAct	BYTE	读写操作：0-写文件，1-读文件，删文件。
byUseRetIndex	BYTE	是否使用上层设置的索引（pRetIndex）：0-不使用，1-使用。
byRes1	BYTE[]	保留。最大长度为 2 字节。
pRetIndex	char*	上层设置的索引，当 byUseRetIndex 值为 0 时，可不设置；byUseRetIndex 值为 1 时设置。
byRes	BYTE[]	保留，最大长度为 56 字节。

A.1.163 NET_EHOME_SS_STORAGE_URI

存储资源的 URI 信息结构体

成员	数据类型	描述
enumType	<i>NET_EHOME_SS_TYPE</i>	[IN]存储服务类型。
szFilename	char[]	[IN]文件标识, 最大长度为 256 字节。
szUri	char[]	[OUT]生成的 URL, 最大长度为 4096 字节 (对应宏定义为“MAX_URL_LEN_SS”)。
byRes	BYTE[]	保留, 最大长度为 64 字节。

A.1.164 NET_EHOME_SS_TOMCAT_MSG

Tomcat 服务器信息结构体

成员	数据类型	描述
szDevUri	char[]	设备请求 URL, 如: “/service/upload_pic.php?puuid=Dvrtest&chan=01&time=2013-07-22%2011:27:01&sequence=01&plateno=A12345&platecolor=0&gpsEW=E&longitude=130345&gpsNS=N&latitude=28650&direction=2898&speed=60&datatype=0&alarmtype=0&zptime=20130409090851000&picnum=2&plateType=0&VehicleColor=0&VehicleType=0&CarType=0&preset=1”, (表 16-13)。最大长度为 4096 字节 (对应宏定义为“MAX_URL_LEN_SS”)。
dwPicNum	DWORD	图片数量。
pPicURLs	char*	图片 URL。
byRes	BYTE[]	保留, 设为 0。最大长度为 64 字节。

表 A-13 URL 中的字段定义

字段	定义
puid	设备注册 ID。
chanCapture	图片抓拍通道号。
time	抓拍时间
sequence	持续抓拍编号。
plateno	车牌号。
platecolor	车牌颜色。
gpsEW	E : 东半球, W : 西半球。
longitude	经度, 计算方式为: 度 \times 3600 \times 100 + 分 \times 60 \times 100 + 秒 \times 100。
gpsNS	N : 北半球, S : 南半球。
latitude	纬度, 计算方式为: 度 \times 3600 \times 100 + 分 \times 60 \times 100 + 秒 \times 100。
direction	方向, 计算方式为: 实际方向值 (北的方向值为 0, 单位: degree) \times 100。
speed	速度, 单位: cm/h。
datatype	数据类型: 0-实时数据, 1-历史数据。
alarmtype	交通违章报警类型: 0-普通车辆通行, 1-黑名单车辆通行。
zptime	抓拍时间格式, “YYYYMMDDHHMMSSXXX” (“年月日时分秒 XXX”), 最后三位为毫秒, 如果不需要精确到毫秒, 最后三位可设为“000”。
picnum	选填, 图片数量, 建议扩展。
plateType	车牌类型。
VehicleColor	车辆颜色。

字段	定义
VehicleType	车辆类型。
CarType	运动目标类型。
pictype	图片类型：0-车牌图片，1-场景图片，2-合成图片。
piclen	可选，图片大小，单位：字节。
picname	选填，图片名称，格式为“年月日时分秒_持续抓拍编号（2-位）_PUID_通道号（2-位）.jpg”。
uploadtype	Uploading mode: 0-（PU）定时抓拍上传至中心，1-（PU）报警抓图上传至中心，2-（CU）手动抓图上传至中心，3-（CU）下载图片，4-（PU）I/O 触发抓图上传至中心。
picid	图片 ID，包含时间，设备 ID 和通道号。
alarm	报警类型：0-硬盘已满报警，1-硬盘出错报警，5-视频丢失报警，6-报警输入报警，7-视频遮盖报警，8-移动侦测报警，10-超速报警，...，0xA00e-ADAS 报警，0xA031-雷达报警。
alarmvideochan	报警视频通道号。
alarmdiskchan	报警硬盘编号。
alarmtime	报警时间。
faceanalysis	选填，是否触发人脸抓拍：0-否，1-是。
facefile	选填，人脸图片文件。
facerect	选填，人脸框。

A.1.165 NET_EHOME_SS_TOMCAT_PARAM

TOMCAT 存储信息结构体

成员	数据类型	描述
byRes	BYTE[]	保留，最大长度为 512 字节。

A.1.166 NET_EHOME_SS_VRB_PARAM

VRB 存储信息结构体

成员	数据类型	描述
byRes	BYTE[]	保留，最大长度为 512 字节。

A.1.167 NET_EHOME_STOPPLAYBACK_PARAM

回放停止请求参数结构体

成员	数据类型	描述
ISessionID	LONG	回放的会话 ID。
IHandle	LONG	输出参数指针，在异步模式中作为异步回调的标识，对应 <u>NET_EHOME_CMSCB_DATA</u> 中的 dwHandle 。
byRes	BYTE[]	保留。最大长度为 120 字节。

A.1.168 NET_EHOME_STOPSTREAM_PARAM

预览停止请求参数结构体

成员	数据类型	描述
ISessionID	LONG	预览的会话 ID, 由 <u>NET_ECMS_StartGetRealStreamV11</u> 或 <u>NET_ECMS_StartGetRealStream</u> 返回。
IHandle	LONG	输出参数指针, 在异步模式中作为异步回调的标识, 对应 <u>NET_EHOME_CMSCB_DATA</u> 中的 dwHandle 。
byRes	BYTE[]	保留。最大长度为 120 字节。

A.1.169 NET_EHOME_STOPVOICETALK_STM_PARAM

通过流媒体服务器的语音对讲停止请求参数结构体

成员	数据类型	描述
ISessionID	LONG	通过流媒体服务器的语音对讲会话 ID。
IHandle	LONG	输出参数句柄, 在异步模式中作为异步回调的标识。对应 <u>NET_EHOME_CMSCB_DATA</u> 中的 dwHandle 。
byRes	BYTE[]	保留。最大长度为 120 字节。

A.1.170 NET_EHOME_SWITCH_DAILY_PLAN

每日定时开关机计划参数结构体

成员	数据类型	描述
dwDayPlanCnt	DWORD	一天中开关机时间节点个数。
struDayPlan	<u>NET_EHOME_SWITCH_PLAN</u> []	开关机参数。最大长度为 16 字节。

A.1.171 NET_EHOME_SWITCH_PLAN

开关机参数结构

成员	数据类型	描述
dwId	DWORD	配置节点编号。
dwSwitchType	DWORD	开关机：1-开机，2-关机。
dwSwitchDate	DWORD	开关机日期。
dwSwitchTime	DWORD	开关机时间，如“235959”表示时间为23:59:59。

A.1.172 NET_EHOME_SWITCH_PLAN_PARAM

发送至终端的开关机计划参数结构体

成员	数据类型	描述
enumPlanType	<u>NET_EHOME_PLAY_SC</u> <u>HEDULE_TYPE1</u>	计划类型：周计划或日计划，参见枚举。
unionSwitchPlan	Union (<u>表 16-14</u>)	计划参数联合体。

表 A-14 计划参数联合体

成员	数据类型	描述
struDailyPlan	<u>NET_EHOME_SWITCH</u> <u>DAILY_PLAN</u>	日计划参数。
struWeeklyPlan	<u>NET_EHOME_SWITCH</u> <u>WEEKLY_PLAN</u>	周计划参数。

A.1.173 NET_EHOME_SWITCH_WEEKLY_PLAN

开关机周计划参数结构体

成员	数据类型	描述
dwWeekPlanCnt	DWORD	一周中配置了开关机计划的天数。
struWeekPlan	<u>NET_EHOME_SWITCHE</u> <u>DAY_OF_WEEK_PLAN</u> []	开关机周计划参数。最大长度为 7 字节。

A.1.174 NET_EHOME_SWITCHE_DAY_OF_WEEK_PLAN

开关机周计划参数结构体

成员	数据类型	描述
dwId	DWORD	周内星期 ID。
enumDayOfWeek	<u>NET_EHOME_DAY_OF</u> <u>WEEK1</u>	星期。
struDayOfWeekPlan	<u>NET_EHOME_SWITCH</u> <u>DAILY_PLAN</u>	开关机日计划参数。

A.1.175 NET_EHOME_SYSCOMPONENT_REDUCED_ADDR

终端组件信息结构体

成员	数据类型	描述
szComponentName	BYTE[]	组件名称，如“SDK”。最大长度为 32 字节。
szComponentPkgName	BYTE[]	组件包名称，如“com.hikvision.sdk”。最大长度为 32 字节。
szComponentVersion	BYTE[]	组件版本信息，如“v2.0.2 build 20180705”。最大长度为 32 字节。
dwIsEnableAutoStart	DWORD	是否开机自启动。
dwIsSupportUpgrade	DWORD	是否支持远程升级。
byRes	BYTE[]	保留。最大长度为 20 字节。

A.1.176 NET_EHOME_TERM_DEFAULT_GATEWAY

终端网关参数结构体

成员	数据类型	描述
szIpAddress	BYTE[]	IPv4 地址的网关。最大长度为 32 字节。
szIpv6Address	BYTE[]	IPv6 地址的网关。最大长度为 32 字节。

A.1.177 NET_EHOME_TERM_IP_ADDRESS

终端 IP 地址参数结构体

成员	数据类型	描述
szIpVersion	BYTE[]	IP 地址版本：v4 和 v6。最大长度为 32 字节。
szIpAddress	BYTE[]	IP 地址。最大长度为 32 字节。
szSubnetMask	BYTE[]	子网掩码。最大长度为 32 字节。

成员	数据类型	描述
szIpv6Address	BYTE[]	IPv6 地址，目前暂不支持。最大长度为 32 字节。
szBitMask	BYTE[]	IPv6 地址的网关。最大长度为 32 字节。
strDefaultGateway	<u>NET_EHOME_TERM_D</u> <u>EFAULT_GATEWAY</u>	终端的网关参数。

A.1.178 NET_EHOME_TERM_TIME_ZONE

终端时区信息结构体

成员	数据类型	描述
szTimeZone	BYTE[]	时区，如 GMT+08。最大长度为 32 字节。

A.1.179 NET_EHOME_TERMINAL_CONTROL

终端控制参数结构体

成员	数据类型	描述
dwControlType	DWORD	控制命令类型，参见枚举 <u>NET_EHOME_CONTROL_TYPE</u> 。
struInsertInfo	<u>NET_EHOME_INSERT_I</u> <u>NFO</u>	插播参数。
struPlanCtrl	<u>NET_EHOME_PLANCTR</u> <u>LPARAM</u>	信息发布终端的定时计划控制参数。
dwConnPort	DWORD	数据传输端口，保留。

A.1.180 NET_EHOME_TERMINAL_CONTROL_V20

终端控制参数结构体 (V20)

成员	数据类型	描述
dwControlType	DWORD	控制命令类型, 参见枚举 <u>NET_EHOME_CONTROL_TYPE</u> 。
struInsertInfo	<u>NET_EHOME_INSERT_I</u> <u>NFO_V20</u>	插播参数。
struPlanCtrl	<u>NET_EHOME_PLANCTR</u> <u>LPARAM</u>	信息发布终端的定时计划控制参数。
dwConnPort	DWORD	数据传输端口, 保留。
byRes	BYTE[]	保留。最大长度为 64 字节。

A.1.181 NET_EHOME_TERMINAL_INFO

终端信息结构体

成员	数据类型	描述
struDevIdentify	<u>NET_EHOME_IDENTIFI</u> <u>CATION</u>	终端 ID。
dwNetUintType	DWORD	网络类型。
byDeviceID	BYTE[]	设备 ID。最大长度为 32 字节 (对应宏定义为 “NET_EHOME_MAX_TYPE_LEN”)。
byPassWord	BYTE[]	密码。最大长度为 32 字节 (对应宏定义为 “NET_EHOME_MAX_TYPE_LEN”)。
byFirmWareVersion	BYTE[]	固件版本信息。最大长度为 32 字节 (对应宏定义为 “NET_EHOME_MAX_TYPE_LEN”)。
byLocalIP	BYTE[]	终端的通信 IP 地址。最大长度为 32 字节 (对应宏定义为 “NET_EHOME_MAX_TYPE_LEN”)。

成员	数据类型	描述
dwLocalPort	DWORD	本地端口号。
dwDevType	DWORD	设备类型。
dwManufacture	DWORD	厂商：0-海康威视。
bDetectPackage	BOOL	封装类型：1-探测包，0-注册包。
bReliableTrans	BOOL	是否支持可靠传输：1-是，0-否。
byServerUserName	BYTE[]	服务器用户名。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_NAME_LEN”）。
byServerUserPassword	BYTE[]	服务器密码。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_NAME_LEN”）。
byTerminalName	BYTE[]	待注册的终端名称。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_NAME_LEN”）。

A.1.182 NET_EHOME_TERMINAL_NAME

终端名称结构体

成员	数据类型	描述
szTermName	BYTE[]	终端名称。最大长度为 100 字节（对应宏定义为“NET_EHOME_MAX_TERM_NAME_LEN”）。

A.1.183 NET_EHOME_TERMINAL_PROGRESS

终端进度信息结构体

成员	数据类型	描述
dwTerminalId	DWORD	终端 ID。
dwTermStatus	DWORD	终端状态：1-正常，2-异常。
dwProgesType	DWORD	进度类型：1-素材进度，2-节目进度，3-日程进度，4-升级进度。
dwUniqueID	DWORD	日程 ID，节目 ID，或素材 ID，升级时该字段无效。
byMainProgress	BYTE	文件下载进度，取值范围从 0 到 100；若取值为 101，代表进度异常。
bySubProgress	BYTE	单个文件的下载进度。

A.1.184 NET_EHOME_TERMINAL_REPORT_INFO

终端上报数据结构体

成员	数据类型	描述
dwSubCmd	DWORD	标识数据内容的类型。
dwContentLen	DWORD	数据内容的长度。
byContentBuf	BYTE[]	储存数据内容的缓冲区。最大长度为 2048 字节。
byRes	BYTE	保留。最大长度为 64 字节。

A.1.185 NET_EHOME_TERMINAL_STATE

终端状态结构体

成员	数据类型	描述
enumPlayState	<i>NET_EHOME_TERMINAL_PLAY_STATE</i>	播放状态。
enumTerminalType	<i>NET_EHOME_TERMINAL_TYPE</i>	终端类型。
enumInsertState	<i>NET_EHOME_INSERT_STATE</i>	插播状态。
bySoftVersion	BYTE[]	终端的软件版本号。最大长度为 32 字节。
byLocalIP	BYTE[]	终端的 IP 地址。最大长度为 32 字节。
dwTermAbnormalState	DWORD	终端是否处于温度异常关机状态：0-否，1-是。

A.1.186 NET_EHOME_TERMINAL_UPG_ST_REPORT

终端返回的升级状态结构体

成员	数据类型	描述
dwLength	DWORD	结构体大小。
dwChecksum	DWORD	校验和。
dwRetVal	DWORD	状态：1000-升级成功，1001-正在升级，1002-升级失败，1003-解码失败，1004-APK 版本不匹配，1005-ROM 版本不匹配。
dwUpgradPercent	DWORD	升级进度（百分比）。

A.1.187 NET_EHOME_TEXT_MESSAGE

文字消息内容结构体

成员	数据类型	描述
dwId	DWORD	文字消息 ID。
szMsgName	BYTE[]	消息名称。最大长度为 64 字节（对应宏定义为“NET_EHOME_MAX_NAME_LEN”）。
szContent	BYTE[]	消息内容。最大长度为 1024 字节（对应宏定义为“NET_EHOME_MAX_TEXT_CONTENT_NUM”）。
szBeginTime	BYTE[]	插播开始时间。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。
szEndTime	BYTE[]	插播结束时间。最大长度为 32 字节（对应宏定义为“NET_EHOME_MAX_TYPE_LEN”）。

A.1.188 NET_EHOME_TIME

时间信息结构体

成员	数据类型	描述
wYear	WORD	年
byMonth	BYTE	月
byDay	BYTE	日
byHour	BYTE	时
byMinute	BYTE	分
bySecond	BYTE	秒
byRes1	BYTE	保留。

成员	数据类型	描述
wMSecond	WORD	毫秒
byRes2	BYTE[]	保留。最大长度为 2 字节。

A.1.189 NET_EHOME_TIME_ADJUST_PARAM

时间同步参数结构体

成员	数据类型	描述
dwTimeZone	DWORD	时区，参见枚举 NET_EHOME_TIME_ZONE 。
szTime	BYTE[]	时间信息，格式为 YYYY-MM-DD HH:MM:SS（年-月-日 时:分:秒）。最大长度为 32 字节。

A.1.190 NET_EHOME_UPGRADE_CONN_PARAM

TCP 链路升级命令参数结构体

成员	数据类型	描述
szServerIP	BYTE[]	服务器 IP 地址。最大长度为 32 字节。
dwServerPort	DWORD	服务器端口号。

A.1.191 NET_EHOME_VERSION_INFO

版本信息结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
sSoftwareVersion	BYTE[]	软件版本号。最大长度为 32 字节（对应宏定义为“MAX_VERSION_LEN”）。
sDSPSoftwareVersion	BYTE[]	编码版本号。最大长度为 32 字节（对应宏定义为“[MAX_VERSION_LEN”）。
sPanelVersion	BYTE[]	面板版本号。最大长度为 32 字节（对应宏定义为“MAX_VERSION_LEN”）。
sHardwareVersion	BYTE[]	硬件版本号。最大长度为 32 字节（对应宏定义为“MAX_VERSION_LEN”）。
byRes	BYTE[]	保留。最大长度为 124 字节。

A.1.192 NET_EHOME_VOICE_TALK_IN

语音对讲请求的输入参数结构体

成员	数据类型	描述
dwVoiceChan	DWORD	语音对讲的通道号。
struStreamSever	<u>NET_EHOME_IPADDRES</u>	流媒体服务器的地址信息。
byEncodingType	<u>NET_EHOME_TALK_ENCODING_TYPE</u> []	语音对讲编码类型。最大长度为 9 字节。
byRes	BYTE[]	保留，设为 0。最大长度为 119 字节。

A.1.193 NET_EHOME_VOICE_TALK_OUT

语音对讲请求的输出参数结构体

成员	数据类型	描述
ISessionID	LONG	语音对讲请求的会话 ID, 由设备返回。
IHandle	LONG	输出参数句柄, 在异步模式中作为异步回调的标识。对应 <i>NET_EHOME_CMSCB_DATA</i> 中的 <i>dwHandle</i> 。
byRes	BYTE[]	保留, 设为 0。最大长度为 124 字节。

A.1.194 NET_EHOME_VOICETALK_DATA

用于转发的语音数据结构体

成员	数据类型	描述
pSendBuf	BYTE*	语音数据缓冲区。
dwDataLen	DWORD	语音数据大小。
byRes	BYTE[]	保留, 设为 0。最大长度为 128 字节。

A.1.195 NET_EHOME_VOICETALK_DATA_CB_INFO

语音对讲回调数据结构体

成员	数据类型	描述
pData	BYTE*	保存语音数据的缓冲区。
dwDataLen	DWORD	保存语音数据的缓冲区大小。
byRes	BYTE[]	保留。最大长度为 128 字节。

A.1.196 NET_EHOME_VOICETALK_DATA_CB_PARAM

语音对讲数据回调参数的结构体

成员	数据类型	描述
fnVoiceTalkDataCB	<u>VOICETALK_DATA_CB</u>	语音对讲数据的回调函数。
pUserData	void*	用户参数。
byRes	BYTE[]	保留，设为 0。最大长度为 128 字节。

A.1.197 NET_EHOME_VOICETALK_NEWLINK_CB_INFO

语音对讲回调信息结构体 (适用于 64 位 Windows 或 Linux 操作系统)

成员	数据类型	描述
szDeviceID	char[]	设备 ID。最大长度为 256 字节 (对应宏定义为 “NET_EHOME_DEVICEID_LEN”)。
dwEncodeType	DWORD	语音编码类型 : 0-G.722, 1-G.711U, 2-G.711A, 3-G.726, 4-AAC, 5-MP2L2, 6-PCM。
sDeviceSerial	char[]	设备序列号。最大长度为 12 字节 (对应宏定义为 “NET_EHOME_SERIAL_LEN”)。
dwAudioChan	DWORD	语音对讲通道号。
lSessionID	LONG	语音对讲的会话 ID, 由设备返回。
byToken	BYTE[]	令牌。最大长度为 64 字节。
fnVoiceTalkDataCB	<u>VOICETALK_DATA_CB</u>	数据回调函数, 其大小为 8 字节。
pUserData	void*	用户参数, 其大小为 8 字节。
byRes	BYTE[]	保留, 设为 0。最大长度为 48 字节。

语音对讲回调信息结构体 (适用于 32 位 Windows 或 Linux 操作系统)

成员	数据类型	描述
szDeviceID	char[]	设备 ID。最大长度为 256 字节 (对应宏定义为 “NET_EHOME_DEVICEID_LEN”)。
dwEncodeType	DWORD	语音编码类型 : 0-G.722, 1-G.711U, 2-G.711A, 3-G.726, 4-AAC, 5-MP2L2, 6-PCM。
sDeviceSerial	char[]	设备序列号。最大长度为 12 字节 (对应宏定义为 “NET_EHOME_SERIAL_LEN”)。
dwAudioChan	DWORD	语音对讲通道号。
lSessionID	LONG	语音对讲的会话 ID, 由设备返回。
byToken	BYTE[]	令牌。最大长度为 64 字节。
fnVoiceTalkDataCB	<u>VOICETALK_DATA_CB</u>	数据回调函数, 其大小为 4 字节。
byRes1	BYTE[]	保留, 最大长度为 4 字节。
pUserData	void*	用户参数, 其大小为 4 字节。
byRes2	BYTE[]	保留, 最大长度为 4 字节。
byRes	BYTE[]	保留, 设为 0。最大长度为 48 字节。

A.1.198 NET_EHOME_VOICETALK_PARA

语音对讲或语音转发参数结构体

成员	数据类型	描述
bNeedCBNoEncData	BOOL	回调的语音类型 : 0-编码后语音, 1-编码前语音 (语音转发时不支持)。
cbVoiceDataCallBack	<u>fVoiceDataCallBack</u>	音频数据的回调函数。

成员	数据类型	描述
dwEncodeType	DWORD	支持的语音编码类型：0-G.722, 1-G.711U, 2-G.711A, 3-G.726, 4-AAC, 5-MP2L2, 6-PCM。
pUser	void*	用户参数。
byVoiceTalk	BYTE	工作模式：0-语音对讲, 1-语音转发。
byDevAudioEnc	BYTE	设备的音频编码类型：0-G.722, 1-G.711U, 2-G.711A, 3-G.726, 4-AAC, 5-MP2L2, 6-PCM。
byRes1	BYTE[]	保留, 设为 0。最大长度为 2 字节。
lHandle	LONG	语音对讲异步回调的消息句柄。
byRes	BYTE[]	保留, 设为 0。最大长度为 56 字节。

A.1.199 NET_EHOME_VOLUME_DAILY_PLAN

音量调整日计划结构体

成员	数据类型	描述
dwDayPlanCnt	DWORD	一天中音量调节的时间节点数量。
struDayPlan	<u>NET_EHOME_VOLUME</u> <u>_PLAN</u>	音量调节日计划参数。最大长度为 8 字节。

A.1.200 NET_EHOME_VOLUME_DAY_OF_WEEK_PLAN

音量调节周计划参数结构体

成员	数据类型	描述
dwId	DWORD	周内星期 ID。
enumDayOfWeek	<u>NET_EHOME_DAY_OF_WEEK1</u>	星期。
struDayOfWeekPlan	<u>NET_EHOME_VOLUME_DAILY_PLAN</u>	音量调节日计划参数。

A.1.201 NET_EHOME_VOLUME_PLAN

音量参数结构体

成员	数据类型	描述
dwId	DWORD	音量调节 ID。
dwVolumeValue	DWORD	音量值。
dwVolumeBeginDate	DWORD	音量调节起始日期。
dwVolumeBeginTime	DWORD	音量调节开始时间，如“235959”表示时间 23:59:59。
dwVolumeEndDate	DWORD	音量调节结束日期。
dwVolumeEndTime	DWORD	音量调节结束时间。

A.1.202 NET_EHOME_VOLUME_PLAN_PARAM

需发送至终端的音量调节参数结构体

成员	数据类型	描述
enumPlanType	<u>NET_EHOME_PLAY_SC</u> <u>HEDULE_TYPE1</u>	计划类型。
unionVolumePlan	Union（表 16-15）	音量调节计划联合体。

表 A-15 音量调节计划联合体

成员	数据类型	描述
struDailyPlan	<u>NET_EHOME_VOLUME</u> <u>DAILY_PLAN</u>	日计划。
struWeeklyPlan	<u>NET_EHOME_VOLUME</u> <u>WEEKLY_PLAN</u>	周计划。

A.1.203 NET_EHOME_VOLUME_WEEKLY_PLAN

音量调节周计划参数结构体

成员	数据类型	描述
dwWeekPlanCnt	DWORD	一周中有音量调节任务的天数。
struWeekPlan	<u>NET_EHOME_VOLUME</u> <u>DAY OF WEEK PLAN</u> []	音量调节周计划参数。最大长度为 7 字节。

A.1.204 NET_EHOME_WEATHER_INFO

天气信息结构体

成员	数据类型	描述
byCityCode	BYTE[]	城市代码。最大长度为 32 字节。
byWeatherData	BYTE[]	天气数据。最大长度为 4*1024 字节。

A.1.205 NET_EHOME_WIRELESS_INFO_CFG

无线参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
dwInfoTransInterval	DWORD	上传时间间隔，单位：秒。
byEnable	BYTE	是否支持：0-否，1-是。
byRes	BYTE[]	保留。最大长度为 47 字节。

A.1.206 NET_EHOME_XML_CFG

传输配置参数结构体

成员	数据类型	描述
pCmdBuf	void*	字符串格式的命令。
dwCmdLen	DWORD	命令的字符数量。
pInBuf	void*	输入参数。
dwInSize	DWORD	输入缓冲区大小。
pOutBuf	void*	输出参数。
dwOutSize	DWORD	输出缓冲区大小。
dwSendTimeOut	DWORD	数据发送超时时间，单位：毫秒，默认：5s。

成员	数据类型	描述
dwRecvTimeOut	DWORD	数据接收超时时间, 单位: 毫秒, 默认: 5s。
pStatusBuf	void*	保存返回状态信息 (XML 格式) 的缓冲区; 若不需要可设为“空”。
dwStatusSize	DWORD	状态缓冲区的大小。
byRes	BYTE[]	保留, 设为 0。最大长度为 24 字节。

相关接口

NET_ECMS_XMLConfig

A.1.207 NET_EHOME_XML_REMOTE_CTRL_PARAM

传输参数结构体

成员	数据类型	描述
dwSize	DWORD	结构体大小。
lpInbuffer	void*	保存输入参数的缓冲区, 由传输的 ISUP 控制命令决定。
dwInBufferSize	DWORD	输入缓冲区大小。
dwSendTimeOut	DWORD	数据发送超时时间, 单位: 毫秒, 默认: 5 s。
dwRecvTimeOut	DWORD	数据接收超时时间, 单位: 毫秒, 默认: 5 s。
lpOutBuffer	void*	输出缓冲区。
dwOutBufferSize	DWORD	输出缓冲区大小。
lpStatusBuffer	void*	状态缓冲区, 若不需要该参数, 可设为“空”。
dwStatusBufferSize	DWORD	状态缓冲区的大小。
byRes	BYTE[]	保留, 设为 0。最大长度为 16 字节。

相关接口

NET_ECMS_XMLRemoteControl

A.1.208 NET_EHOME_ZONE

区域参数结构体

成员	数据类型	描述
dwX	DWORD	X 轴坐标。
dwY	DWORD	Y 轴坐标。
dwWidth	DWORD	区域宽度。
dwHeight	DWORD	区域高度。

A.2 枚举定义

A.2.1 EN_ALARM_TYPE

枚举报警类型

枚举定义

```
typedef enum{
    ALARM_TYPE_DISK_FULL           = 0,
    ALARM_TYPE_DISK_WRError       = 1,
    ALARM_TYPE_VIDEO_LOST         = 5,
    ALARM_TYPE_EXTERNAL           = 6,
    ALARM_TYPE_VIDEO_COVERED      = 7,
    ALARM_TYPE_MOTION             = 8,
    ALARM_TYPE_STANDARD_NOTMATCH  = 9,
    ALARM_TYPE_SPEEDLIMIT_EXCEED = 10,
    ALARM_TYPE_PIR                = 11,
    ALARM_TYPE_WIRELESS           = 12,
    ALARM_TYPE_CALL_HELP          = 13,
    ALARM_TYPE_DISARM             = 14,
    ALARM_TYPE_STREAM_PRIVATE     = 15,
    ALARM_TYPE_PIC_UPLOAD_FAIL    = 16,
    ALARM_TYPE_LOCAL_REC_EXCEPTION = 17,
    ALARM_TYPE_UPGRADE_FAIL       = 18,
    ALARM_TYPE_ILLEGAL_ACCESS     = 19,
```

```
ALARM_TYPE_SOUNDLIMIT_EXCEED = 80,
ALARM_TYPE_TRIFFIC_VIOLATION = 90,
ALARM_TYPE_ALARM_CONTROL = 96,
ALARM_TYPE_FACE_DETECTION = 97,
ALARM_TYPE_DEFOUSE_DETECTION = 98,
ALARM_TYPE_AUDIO_EXCEPTION = 99,
ALARM_TYPE_SCENE_CHANGE = 100,
ALARM_TYPE_TRAVERSE_PLANE = 101,
ALARM_TYPE_ENTER_AREA = 102,
ALARM_TYPE_LEAVE_AREA = 103,
ALARM_TYPE_INTRUSION = 104,
ALARM_TYPE_LOITER = 105,
ALARM_TYPE_LEFT_TAKE = 106,
ALARM_TYPE_CAR_STOP = 107,
ALARM_TYPE_MOVE_FAST = 108,
ALARM_TYPE_HIGH_DENSITY = 109,
ALARM_TYPE_PDC_BY_TIME = 110,
ALARM_TYPE_PDC_BY_FRAME = 111,
ALARM_TYPE_LEFT = 112,
ALARM_TYPE_TAKE = 113,
ALARM_TYPE_ROLLOVER = 114,
ALARM_TYPE_COLLISION = 115,
ALARM_TYPE_FLOW_OVERRUN = 256,
ALARM_TYPE_WARN_FLOW_OVERRUN = 257,
ALARM_TYPE_DEV_CHANGED_STATUS = 700,
ALARM_TYPE_CHAN_CHANGED_STATUS = 701,
ALARM_TYPE_HD_CHANGED_STATUS = 702,
ALARM_TYPE_DEV_TIMING_STATUS = 703,
ALARM_TYPE_CHAN_TIMING_STATUS = 704,
ALARM_TYPE_HD_TIMING_STATUS = 705,
ALARM_TYPE_RECORD_ABNORMAL = 706,
ALARM_TYPE_ENV_LIMIT = 8800,
ALARM_TYPE_ENV_REAL_TIME = 8801,
ALARM_TYPE_ENV_EXCEPTION = 8802,
ALARM_TYPE_HIGH_TEMP = 40961,
ALARM_TYPE_ACC_EXCEPTION = 40962,
ALARM_TYPE_RAPID_ACCELERATION = 40963,
ALARM_TYPE_RAPID_DECELERATION = 40964,
ALARM_TYPE_COLLISION_V40 = 40965,
ALARM_TYPE_ROLLOVER_V40 = 40966,
ALARM_TYPE_RAPID_TURN_LEFT = 40967,
ALARM_TYPE_RAPID_TURN_RIGHT = 40968,
ALARM_TYPE_ABNORMAL_DRIVING_BEHAVIOR = 40969,
ALARM_TYPE_OVERLOAD = 40970,
ALARM_TYPE_LEFT_CROSS_LINE = 40971,
ALARM_TYPE_RIGHT_CROSS_LINE = 40972,
ALARM_TYPE_OPEN_DOOR_WITH_SPEED = 40973,
ALARM_TYPE_ADAS = 40974,
ALARM_TYPE_RADAR = 41009
}EN_ALARM_TYPE;
```

成员

ALARM_TYPE_DISK_FULL

硬盘已满报警

ALARM_TYPE_DISK_WRERROR

硬盘读写出错报警

ALARM_TYPE_VIDEO_LOST

视频（信号）丢失报警

ALARM_TYPE_EXTERNAL

外部（信号量）报警

ALARM_TYPE_VIDEO_COVERED

视频遮盖报警

ALARM_TYPE_MOTION

移动侦测

ALARM_TYPE_STANDARD_NOTMATCH

视频制式不匹配报警

ALARM_TYPE_SPEEDLIMIT_EXCEED

超速报警

ALARM_TYPE_PIR

PIR 报警

ALARM_TYPE_WIRELESS

无线报警

ALARM_TYPE_CALL_HELP

呼救报警

ALARM_TYPE_DISARM

布撤防报警

ALARM_TYPE_STREAM_PRIVATE

码流隐私状态改变报警

ALARM_TYPE_PIC_UPLOAD_FAIL

设备上传图片失败报警

ALARM_TYPE_LOCAL_REC_EXCEPTION

设备本地录像（取证）异常报警

ALARM_TYPE_UPGRADE_FAIL

设备版本升级失败报警

ALARM_TYPE_ILLEGAL_ACCESS

非法访问报警

ALARM_TYPE_SOUNDLIMIT_EXCEED

声音分贝数超标报警

ALARM_TYPE_TRIFFIC_VIOLATION

违章报警

ALARM_TYPE_ALARM_CONTROL

布防报警

ALARM_TYPE_FACE_DETECTION

人脸侦测报警

ALARM_TYPE_DEFOUSE_DETECTION

虚焦侦测报警

ALARM_TYPE_AUDIO_EXCEPTION

音频异常报警

ALARM_TYPE_SCENE_CHANGE

场景变更侦测报警

ALARM_TYPE_TRAVERSE_PLANE

越界侦测报警

ALARM_TYPE_ENTER_AREA

进入区域侦测报警

ALARM_TYPE_LEAVE_AREA

离开区域侦测报警

ALARM_TYPE_INTRUSION

区域入侵侦测报警

ALARM_TYPE_LOITER

徘徊侦测报警

ALARM_TYPE_LEFT_TAKE

遗留物品拿取侦测报警

ALARM_TYPE_CAR_STOP

停车侦测报警

ALARM_TYPE_MOVE_FAST

快速移动侦测报警

ALARM_TYPE_HIGH_DENSITY

人员聚集侦测报警

ALARM_TYPE_PDC_BY_TIME

按时间段统计客流量报警

ALARM_TYPE_PDC_BY_FRAME

单帧统计客流量报警

ALARM_TYPE_LEFT

物品遗留侦测报警

ALARM_TYPE_TAKE

物品拿取侦测报警

ALARM_TYPE_ROLLOVER

侧翻报警

ALARM_TYPE_COLLISION

碰撞报警

ALARM_TYPE_FLOW_OVERRUN

流量超限报警

ALARM_TYPE_WARN_FLOW_OVERRUN

人员超限提醒

ALARM_TYPE_DEV_CHANGED_STATUS

设备状态改变报警

ALARM_TYPE_CHAN_CHANGED_STATUS

通道状态改变报警

ALARM_TYPE_HD_CHANGED_STATUS

硬盘状态改变报警

ALARM_TYPE_DEV_TIMING_STATUS

定时上传设备状态报警

ALARM_TYPE_CHAN_TIMING_STATUS

定时上传通道状态报警

ALARM_TYPE_HD_TIMING_STATUS

定时上传硬盘状态报警

ALARM_TYPE_RECORD_ABNORMAL

录像异常报警

ALARM_TYPE_ENV_LIMIT

动环环境量超限报警

ALARM_TYPE_ENV_REAL_TIME

动环环境量实时数据上传报警

ALARM_TYPE_ENV_EXCEPTION

动环环境量异常上传报警

ALARM_TYPE_HIGH_TEMP

温度过高报警

ALARM_TYPE_ACC_EXCEPTION

加速异常报警

ALARM_TYPE_RAPID_ACCELERATION

急加速报警

ALARM_TYPE_RAPID_DECELERATION

急减速报警

ALARM_TYPE_COLLISION_V40

碰撞报警

ALARM_TYPE_ROLLOVER_V40

侧翻报警

ALARM_TYPE_RAPID_TURN_LEFT

急左转弯报警

ALARM_TYPE_RAPID_TURN_RIGHT

急右转弯报警

ALARM_TYPE_ABNORMAL_DRIVING_BEHAVIOR

异常驾驶行为报警

ALARM_TYPE_OVERLOAD

超载报警

ALARM_TYPE_LEFT_CROSS_LINE

左压线报警

ALARM_TYPE_RIGHT_CROSS_LINE

右压线报警

ALARM_TYPE_OPEN_DOOR_WITH_SPEED

带速开门报警

ALARM_TYPE_ADAS

主动安全（ADAS）报警

ALARM_TYPE_RADAR

雷达报警

A.2.2 LONG_LINK_MSG

枚举长连接类型

枚举定义

```
typedef enum{
    LONG_CFG_CREATED           =0,
    LONG_CFG_CREATE_FAIL      =1,
    LONG_CFG_DATA              =2,
    LONG_CFG_TERMINATE         =3
}LONG_LINK_MSG;
```

成员

LONG_CFG_CREATED

长连接建立成功。

LONG_CFG_CREATE_FAIL

长连接建立失败。

LONG_CFG_DATA

长连接普通数据。

LONG_CFG_TERMINATE

销毁长连接。

A.2.3 NET_CMS_ENUM_PROXY_TYPE

枚举代理类型

枚举定义

```
typedef enum tagNET_CMS_ENUM_PROXY_TYPE{
    ENUM_PROXY_TYPE_NETSDK    = 0,
    ENUM_PROXY_TYPE_HTTP      = 1
}NET_CMS_ENUM_PROXY_TYPE;
```

成员

ENUM_PROXY_TYPE_NETSDK

海康威视私有协议代理。

ENUM_PROXY_TYPE_HTTP

HTTP 代理。

A.2.4 NET_EHOME_ADDRESS_TYPE

枚举地址类型。

枚举定义

```
typedef enum{
    NET_EHOME_HOST_MODE,
    NET_EHOME_IP_MODE,
}NET_EHOME_ADDRESS_TYPE;
```

成员

NET_EHOME_HOST_MODE

域名。

NET_EHOME_IP_MODE

IP 地址。

A.2.5 NET_EHOME_ADDRFORMAT_CMD

枚举地址类型。

枚举定义

```
typedef enum{
    ADDRTYPE_IPV4 = 0,
    ADDRTYPE_IPV6 = 1,
    HOSTNAME      = 2,
}NET_EHOME_ADDRFORMAT_CMD;
```

成员

ADDRTYPE_IPV4

IPv4 地址。

ADDRTYPE_IPV6

IPv6 地址。

HOSTNAME

域名。

A.2.6 NET_EHOME_CALLBACK_TYPE

枚举回调类型。

枚举定义

```
typedef enum tagNET_EHOME_CALLBACK_TYPE{
    NET_EHOME_REALSTREAM_CB    = 0,
    NET_EHOME_PLAYBACK_CB     = 1,
    NET_EHOME_VOICETALK_CB    = 2,
    NET_EHOME_PASSTHROUGH_CB  = 3,
    NET_EHOME_V2_VOICETALK_CB = 4
}NET_EHOME_CALLBACK_TYPE;
```

成员

NET_EHOME_REALSTREAM_CB

预览异步回调，对应 [NET_ECMS_StartGetRealStream](#)、[NET_ECMS_StartGetRealStreamV11](#)、[NET_ECMS_StartPushRealStream](#)、[NET_ECMS_StopGetRealStream](#)、[NET_ECMS_StopGetRealStreamEx](#) 接口的异步返回。

NET_EHOME_PLAYBACK_CB

回放异步回调，对应 [NET_ECMS_StartPlayBack](#)、[NET_ECMS_StopPlayBack](#)、[NET_ECMS_StopPlayBackEx](#)、[NET_ECMS_StartPushPlayBack](#) [NET_ECMS_StartPushPlayBack](#)、[NET_ECMS_PlayBackOperate](#) 接口的异步返回。

NET_EHOME_VOICETALK_CB

语音（对讲）转发异步回调。对应 [NET_ECMS_StartVoiceWithStmServer](#)、[NET_ECMS_StartPushVoiceStream](#)、[NET_ECMS_StopVoiceTalkWithStmServer](#)、[NET_ECMS_StopVoiceTalkWithStmServerEx](#) 接口的异步返回。

NET_EHOME_PASSTHROUGH_CB

ISAPI 透传异步回调。对应 NET_ECMS_ISAPIPassThrough 接口的异步返回。

NET_EHOME_V2_VOICETALK_CB

ISUP2.0 协议语音对讲异步回调。对应 NET_ECMS_StartVoiceTalk 接口的异步返回。

A.2.7 NET_EHOME_CMS_INIT_CFG_TYPE

枚举初始化配置类型。

枚举定义

```
typedef enum tagNET_EHOME_CMS_INIT_CFG_TYPE{
    NET_EHOME_CMS_INIT_CFG_LIBEAY_PATH = 0,
    NET_EHOME_CMS_INIT_CFG_SSLEAY_PATH = 1
}NET_EHOME_CMS_INIT_CFG_TYPE;
```

成员

NET_EHOME_CMS_INIT_CFG_LIBEAY_PATH

设置 OpenSSL 的加密库 (libeay32.dll or libcrypto.so) 所在路径，路径为包含库名的完整路径。

NET_EHOME_CMS_INIT_CFG_SSLEAY_PATH

设置 OpenSSL 的通信库 (ssleay32.dll or libssl.so) 所在路径，路径为包含库名的完整路径。

A.2.8 NET_EHOME_CONTROL_TYPE

枚举终端控制命令

枚举定义

```
enum NET_EHOME_CONTROL_TYPE{
    NET_EHOME_CONTROL_STARTPLAY = 1,
    NET_EHOME_CONTROL_STOPPLAY,
    NET_EHOME_CONTROL_INSERT,
    NET_EHOME_CONTROL_STOPINSERT,
    NET_EHOME_CONTROL_POWERON,
    NET_EHOME_CONTROL_POWEROFF,
    NET_EHOME_CONTROL_REBOOT,
    NET_EHOME_CONTROL_RESTORECONFIG,
    NET_EHOME_CONTROL_SCREENOPEN,
    NET_EHOME_CONTROL_SCREENCLOSE,
    NET_EHOME_CONTROL_PLANCTRL,
    NET_EHOME_CONTROL_CANCELINSERT,
    NET_EHOME_CONTROL_PUBLISH_STATE_SHOW,
```

```
NET_EHOME_CONTROL_PUBLISH_STATE_HIDE,  
};
```

成员

NET_EHOME_CONTROL_STARTPLAY

开始播放。

NET_EHOME_CONTROL_STOPPLAY

停止播放。

NET_EHOME_CONTROL_INSERT

开始插播。

NET_EHOME_CONTROL_STOPINSERT

停止插播。

NET_EHOME_CONTROL_POWERON

开机。

NET_EHOME_CONTROL_POWEROFF

关机。

NET_EHOME_CONTROL_REBOOT

重启。

NET_EHOME_CONTROL_RESTORECONFIG

回复默认设置。

NET_EHOME_CONTROL_SCREENOPEN

屏幕开。

NET_EHOME_CONTROL_SCREENCLOSE

屏幕关。

NET_EHOME_CONTROL_PLANCTRL

定时控制音量和开关机计划。

NET_EHOME_CONTROL_CANCELINSERT

取消插播。

NET_EHOME_CONTROL_PUBLISH_STATE_SHOW

显示发布进度。

NET_EHOME_CONTROL_PUBLISH_STATE_HIDE

隐藏发布进度。

A.2.9 NET_EHOME_DAY_OF_WEEK1

枚举一周中的星期。

枚举定义

```
typedef enum{
    INVALID_DAY = 0,
    MONDAY      = 1,
    TUESDAY     = 2,
    WEDNESDAY   = 3,
    THURSDAY    = 4,
    FRIDAY      = 5,
    SATURDAY    = 6,
    SUNDAY      = 7,
}NET_EHOME_DAY_OF_WEEK1;
```

成员

INVALID_DAY

无效

MONDAY

星期一

TUESDAY

星期二

WEDNESDAY

星期三

THURSDAY

星期四

FRIDAY

星期五

SATURDAY

星期六

SUNDAY

星期日

A.2.10 NET_EHOME_EALARM_INIT_CFG_TYPE

枚举报警管理服务器的初始化配置类型。

枚举定义

```
typedef enum tagNET_EHOME_EALARM_INIT_CFG_TYPE{
    NET_EHOME_EALARM_INIT_CFG_LIBEAY_PATH = 0,
    NET_EHOME_EALARM_INIT_CFG_SSLEAY_PATH = 1
}NET_EHOME_EALARM_INIT_CFG_TYPE;
```

成员

NET_EHOME_EALARM_INIT_CFG_LIBEAY_PATH

设置 OpenSSL 的加密库（libeay32.dll or libcrypto.so）所在路径，路径为包含库名的完整路径。

NET_EHOME_EALARM_INIT_CFG_SSLEAY_PATH

设置 OpenSSL 的通信库（ssleay32.dll or libssl.so）所在路径，路径为包含库名的完整路径。

A.2.11 NET_EHOME_ERROR_CODE

枚举服务器和终端定义的错误类型。

枚举定义

```
enum NET_EHOME_ERROR_CODE{
    EHOME_OK = 0,
    EHOME_PASSERWORD_ERROR = 1,
    EHOME_NOT_SUPPORT = 2,
    EHOME_USERNAME_ERROR = 3,
    EHOME_USERNAME_PASSWORD_ERROR = 4,
    EHOME_TERM_RNAME_REPEAT = 5,
    EHOME_TERM_SERIAL_REPEAT = 6,
    EHOME_PARAM_ERROR = 7,
};
```

成员

EHOME_OK

没有错误。

EHOME_PASSERWORD_ERROR

密码错误。

EHOME_NOT_SUPPORT

终端设备不支持。

EHOME_USERNAME_ERROR

用户名错误。

EHOME_USERNAME_PASSWORD_ERROR

用户名和密码错误。

EHOME_TERM_RNAME_REPEAT

终端名称重复。

EHOME_TERM_SERIAL_REPEAT

终端序列号重复。

EHOME_PARAM_ERROR

参数错误。

A.2.12 NET_EHOME_ESTREAM_INIT_CFG_TYPE

枚举流媒体服务器的初始化配置类型。

枚举定义

```
typedef enum tagNET_EHOME_ESTREAM_INIT_CFG_TYPE{
    NET_EHOME_ESTREAM_INIT_CFG_LIBEAY_PATH           = 0,
    NET_EHOME_ESTREAM_INIT_CFG_SSLEAY_PATH           = 1,
    NET_EHOME_ESTREAM_INIT_CFG_USERCERTIFICATE_PATH  = 2,
    NET_EHOME_ESTREAM_INIT_CFG_USERPRIVATEKEY_PATH  = 3
}NET_EHOME_ESTREAM_INIT_CFG_TYPE;
```

成员

NET_EHOME_ESTREAM_INIT_CFG_LIBEAY_PATH

设置 OpenSSL 的加密库(i.e., libeay32.dll or libcrypto.so)的路径，路径为包含库名的完整路径。

NET_EHOME_ESTREAM_INIT_CFG_SSLEAY_PATH

设置 OpenSSL 的通信库(i.e., ssleay32.dll or libssl.so)的路径，路径为包含库名的完整路径。

NET_EHOME_ESTREAM_INIT_CFG_USERCERTIFICATE_PATH

设置 TLS 监听和 DTLS 监听的服务器证书路径（TLS 和 DTLS 复用同一个证书）。未设置加密证书，回放、预览、语音转发 TCP 加密业务中，推流接口会报错。

NET_EHOME_ESTREAM_INIT_CFG_USERPRIVATEKEY_PATH

设置 TLS 监听和 DTLS 监听的私钥证书路径（TLS 和 DTLS 复用同一个证书）。未设置加密证书，回放、预览、语音转发 TCP 加密业务中，推流接口会报错。

A.2.13 NET_EHOME_INSERT_STATE

枚举插播状态。

枚举定义

```
typedef enum{
    NET_EHOME_INVALID_INSERT_TYPE = 0,
    NET_EHOME_INSERT_END          = 1,
    NET_EHOME_INSERT_PUBLISH     = 2,
    NET_EHOME_INSERT_START       = 3,
    NET_EHOME_INSERT_FAIL        = 4,
}NET_EHOME_INSERT_STATE;
```

成员

NET_EHOME_INVALID_INSERT_TYPE

无效类型。

NET_EHOME_INSERT_END

插播结束。

NET_EHOME_INSERT_PUBLISH

向终端发送数据包。

NET_EHOME_INSERT_START

插播开始。

NET_EHOME_INSERT_FAIL

插播失败。

A.2.14 NET_EHOME_IPC_TYPE

枚举网络摄像机的类型。

枚举定义

```
typedef enum{
    NET_EHOME_NORMAL_IPC,
    NET_EHOME_SMART_PASSENGER_IPC,
    NET_EHOME_SMART_VALID_PASSENGER_IPC,
}NET_EHOME_IPC_TYPE;
```

成员

NET_EHOME_NORMAL_IPC

普通网络摄像机。

NET_EHOME_SMART_PASSENGER_IPC

客流统计网络摄像机。

NET_EHOME_SMART_VALID_PASSENGER_IPC

人脸识别网络摄像机。

A.2.15 NET_EHOME_LOCAL_CFG_TYPE

枚举本地配置类型。

枚举定义

```
enum NET_EHOME_LOCAL_CFG_TYPE{
    UNDEFINE                = -1,
    ACTIVE_ACCESS_SECURITY  = 0,
    AMS_ADDRESS              = 1,
    SEND_PARAM               = 2,
    SET_REREGISTER_MODE     = 3,
    LOCAL_CFG_TYPE_GENERAL  = 4,
    COM_PATH                 = 5,
    SESSIONKEY_REQ_MOD      = 6,
    DEV_DAS_PINGREQ_CALLBACK = 7,
    REGISTER_LISTEN_MODE    = 8,
    STREAM_PLAYBACK_PARAM   = 9
}NET_EHOME_LOCAL_CFG_TYPE, *LPNET_EHOME_LOCAL_CFG_TYPE;
```

成员

UNDEFINE

未定义。

ACTIVE_ACCESS_SECURITY

设置用于设备连接到服务器的安全参数。

AMS_ADDRESS

设置报警管理服务器（AMS）的本地环回地址信息。

SEND_PARAM

设置发送参数。

SET_REREGISTER_MODE

设置设备的重复注册模式。

LOCAL_CFG_TYPE_GENERAL

设置通用参数。

COM_PATH

配置 COM 路径。

SESSIONKEY_REQ_MOD

设置会话密钥请求的回调状态。

DEV_DAS_PINGREQ_CALLBACK

设备心跳注册回调。

REGISTER_LISTEN_MODE

注册监听模式，对应结构体为 NET_EHOME_REGISTER_LISTEN_MODE。

STREAM_PLAYBACK_PARAM

回放本地参数配置。

A.2.16 NET_EHOME_PLANCTRL_TYPE

枚举控制的类型。

枚举定义

```
enum NET_EHOME_PLANCTRL_TYPE{
    NET_EHOME_CONTROL_SWITCHPLAY = 1,
    NET_EHOME_CONTROL_VOLUMEPLAY = 2,
    NET_EHOME_CONTROL_INPUTPLAY = 3,
};
```

成员

NET_EHOME_CONTROL_SWITCHPLAY

定时开关机计划。

NET_EHOME_CONTROL_VOLUMEPLAY

音量调节。

NET_EHOME_CONTROL_INPUTPLAY

是否启用定时输入计划。

A.2.17 NET_EHOME_PLAY_SCHEDULE_TYPE1

枚举计划类型。

枚举定义

```
typedef enum{
    INVALID_PLAN      = 0,
    DAYLY_PLAN        = 1,
    WEEKLY_PLAN       = 2,
    SELF_DEF_PLAN     = 3,
    CIRCLE_PLAN       = 4,
}NET_EHOME_PLAY_SCHEDULE_TYPE1;
```

成员

INVALID_PLAN

无效的计划类型。

DAYLY_PLAN

日计划。

WEEKLY_PLAN

周计划。

SELF_DEF_PLAN

自定义计划。

CIRCLE_PLAN

轮播计划。

A.2.18 NET_EHOME_PLAYBACK_OPERATE_MODE

枚举控件操作类型。

枚举定义

```
typedef enum tagENUM_NET_EHOME_PLAYBACK_OPERATE_MODE{
    PLAYBACK_OPERATE_UNKNOW = -1,
    PLAYBACK_OPERATE_PAUSE  = 0,
    PLAYBACK_OPERATE_RESTART = 1
}ENUM_NET_EHOME_PLAYBACK_OPERATE_MODE;
```

成员

PLAYBACK_OPERATE_UNKNOW

无效的控件操作类型。

PLAYBACK_OPERATE_PAUSE

暂停回放。

PLAYBACK_OPERATE_RESTART

恢复回放。

A.2.19 NET_EHOME_POS_MODE

枚举坐标类型。

枚举定义

```
enum NET_EHOME_POS_MODE{
    NET_EHOME_POS_MODE_ILLEGAL    = 0,
    NET_EHOME_POS_MODE_STANDARD   = 1,
    NET_EHOME_POS_MODE_RESOLUTION = 2,
};
```

成员

NET_EHOME_POS_MODE_ILLEGAL

无效的坐标类型。

NET_EHOME_POS_MODE_STANDARD

基准坐标，如 1920 × 1920。

NET_EHOME_POS_MODE_RESOLUTION

分辨率坐标。

A.2.20 NET_EHOME_POWER_ONOFF

枚举开关机状态。

枚举定义

```
typedef enum{
    POWER_ON    = 1,
    POWER_OFF   = 2,
}NET_EHOME_POWER_ONOFF;
```

成员

POWER_ON

开机。

POWER_OFF

关机。

A.2.21 NET_EHOME_PROGRAM_INSERT_TYPE

枚举插播类型。

枚举定义

```
enum NET_EHOME_PROGRAM_INSERT_TYPE{
    NET_EHOME_INSERT_TYPE_MATERIAL = 1,
    NET_EHOME_INSERT_TYPE_PROGRAM = 2,
};
```

成员

NET_EHOME_INSERT_TYPE_MATERIAL

素材。

NET_EHOME_INSERT_TYPE_PROGRAM

节目。

A.2.22 NET_EHOME_REFRESH_TYPE

枚举数据刷新类型。

枚举定义

```
typedef enum{
    NET_EHOME_SINGLE_REFRESH_TYPE = 1,
    NET_EHOME_WHOLE_REFRESH_TYPE = 2,
    NET_EHOME_OTHER_REFRESH_TYPE = 3,
}NET_EHOME_REFRESH_TYPE;
```

成员

NET_EHOME_SINGLE_REFRESH_TYPE

刷新单个数据。

NET_EHOME_WHOLE_REFRESH_TYPE

刷新所有数据。

NET_EHOME_OTHER_REFRESH_TYPE

其他。

A.2.23 NET_EHOME_REGISTER_TYPE

枚举注册回调类型。

枚举定义

```
typedef enum tagNET_EHOME_REGISTER_TYPE{
    ENUM_UNKNOWN          =-1,
    ENUM_DEV_ON           =0,
    ENUM_DEV_OFF          =1,
    ENUM_DEV_ADDRESS_CHANGED =2,
    ENUM_EHOME50_DEV_AUTH =3,
    ENUM_EHOME50_DEV_SESSIONKEY =4,
    ENUM_EHOME50_DEV_DAS_REQ =5,
    ENUM_DEV_SESSIONKEY_REQ =6,
    ENUM_DEV_DAS_REREGISTER =7,
    ENUM_DEV_DAS_PINGREQ   =8,
    ENUM_DEV_DAS_EHOMEKEY_ERROR =9,
    ENUM_DEV_SESSIONKEY_ERROR =10,
    ENUM_DEV_SLEEP        =11
}NET_EHOME_REGISTER_TYPE
```

成员

ENUM_UNKNOWN

未知信息。

ENUM_DEV_ON

设备在线信息。

ENUM_DEV_OFF

设备离线信息。

ENUM_DEV_ADDRESS_CHANGED

设备地址改变信息。

ENUM_EHOME50_DEV_AUTH

支持 5.0 版本 ISUP 设备的认证信息。

ENUM_EHOME50_DEV_SESSIONKEY

支持 5.0 版本 ISUP 设备的会话密钥信息。

ENUM_EHOME50_DEV_DAS_REQ

支持 5.0 版本 ISUP 设备的重定向请求信息。

ENUM_DEV_SESSIONKEY_REQ

支持 5.0 版本 ISUP 设备的 SessionKey 请求信息。

ENUM_DEV_DAS_REREGISTER

设备的重注册信息。

ENUM_DEV_DAS_PINGREQ

设备注册心跳。

ENUM_DEV_DAS_EHOMEKEY_ERROR

校验密码失败信息。

ENUM_DEV_SESSIONKEY_ERROR

Sessionkey 交互异常。

ENUM_DEV_SLEEP

设备进入休眠状态

说明

休眠状态下，设备无法对预览、回放、语音对讲、配置等请求作响应；可调用接口 `NET_ECMS_WakeUp` 唤醒设备

See Also

[NET_ECMS_WakeUp](#)

A.2.24 NET_EHOME_SS_CLIENT_TYPE

枚举文件上传客户端类型。

枚举定义

```
enum NET_EHOME_SS_CLIENT_TYPE (  
    NET_EHOME_SS_CLIENT_TYPE_TOMCAT = 1,  
    NET_EHOME_SS_CLIENT_TYPE_VRB    = 2,  
    NET_EHOME_SS_CLIENT_TYPE_KMS    = 3,  
    NET_EHOME_SS_CLIENT_TYPE_CLOUD  = 4,  
    NET_EHOME_SS_CLIENT_TYPE_CENTRAL= 5  
)NET_EHOME_SS_CLIENT_TYPE
```

成员

NET_EHOME_SS_CLIENT_TYPE_TOMCAT

支持 Tomcat 协议的客户端

NET_EHOME_SS_CLIENT_TYPE_VRB

支持 VRB 协议的客户端

NET_EHOME_SS_CLIENT_TYPE_KMS

支持 KMS 协议的客户端

NET_EHOME_SS_CLIENT_TYPE_CLOUD

支持 ISUP5.0 的客户端

NET_EHOME_SS_CLIENT_TYPE_CENTRAL

中心存储流式存储协议

A.2.25 NET_EHOME_SS_INIT_CFG_TYPE

枚举初始化配置类型。

枚举定义

```
enum NET_EHOME_SS_INIT_CFG_TYPE{
    NET_EHOME_SS_INIT_CFG_SDK_PATH           = 1,
    NET_EHOME_SS_INIT_CFG_CLOUD_TIME_DIFF    = 2,
    NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT     = 3,
    NET_EHOME_SS_INIT_CFG_LIBEAY_PATH        = 4,
    NET_EHOME_SS_INIT_CFG_SSLEAY_PATH        = 5,
    NET_EHOME_SS_INIT_CFG_SQLITE3_PATH       = 6
}NET_EHOME_SS_INIT_CFG_TYPE
```

Members

NET_EHOME_SS_INIT_CFG_SDK_PATH

设置存储管理模块的加载路径（只对 Linux 操作系统有效）

NET_EHOME_SS_INIT_CFG_CLOUD_TIME_DIFF

设置云存储请求的时间差，单位：分钟。其值范围为 5 到 60，默认值为 15。

NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT

设置公共网络的地址（只有在私网和公网映射时有效）。

NET_EHOME_SS_INIT_CFG_LIBEAY_PATH

设置 OpenSSL 的加密库(i.e., libeay32.dll or libcrypto.so)的路径，路径为包含库名的完整路径。

NET_EHOME_SS_INIT_CFG_SSLEAY_PATH

设置 OpenSSL 的通信库(i.e., ssleay32.dll or libssl.so)的路径，路径为包含库名的完整路径。

NET_EHOME_SS_INIT_CFG_SQLITE3_PATH

设置 sqlite3 库（Windows 下为 sqlite3.dll，Linux 下为 libsqlite3.so）的路径。

A.2.26 NET_EHOME_SS_MSG_TYPE

枚举存储服务器的回调数据类型。

枚举定义

```
enum _NET_EHOME_SS_MSG_TYPE_{
    NET_EHOME_SS_MSG_TOMCAT      = 1,
    NET_EHOME_SS_MSG_KMS_USER_PWD = 2,
    NET_EHOME_SS_MSG_CLOUD_AK    = 3
};
```

成员

NET_EHOME_SS_MSG_TOMCAT

回调 Tomcat 服务器信息，详情参见结构体 [NET_EHOME_SS_TOMCAT_MSG](#)。

NET_EHOME_SS_MSG_KMS_USER_PWD

回调密钥管理服务器（KMS）的用户名和密码。

NET_EHOME_SS_MSG_CLOUD_AK

回调 5.0 版本 ISUP 设备的访问密钥。

A.2.27 NET_EHOME_SS_TYPE

枚举存储服务类型。

枚举定义

```
enum NET_EHOME_SS_TYPE{
    NET_EHOME_SS_TYPE_TOMCAT = 1,
    NET_EHOME_SS_TYPE_VRB,   = 2
    NET_EHOME_SS_TYPE_KMS,   = 3
    NET_EHOME_SS_TYPE_CLOUD  = 4
}NET_EHOME_SS_TYPE;
```

成员

NET_EHOME_SS_TYPE_TOMCAT

Tomcat

NET_EHOME_SS_TYPE_VRB

VRB

NET_EHOME_SS_TYPE_KMS

KMS

NET_EHOME_SS_TYPE_CLOUD

ISUP5.0 云存储协议

A.2.28 NET_EHOME_STREAM_TYPE

枚举码流类型。

枚举定义

```
typedef enum{
    NET_EHOME_MAIN    = 1,
    NET_EHOME_SUB     = 2,
    NET_EHOME_THIRD   = 3,
}NET_EHOME_STREAM_TYPE;
```

成员

NET_EHOME_MAIN

主码流。

NET_EHOME_SUB

子码流。

NET_EHOME_THIRD

第三码流。

A.2.29 NET_EHOME_TALK_ENCODING_TYPE

语音对讲编码类型。

枚举定义

```
typedef enum tagNET_EHOME_TALK_ENCODING_TYPE{
    ENUM_ENCODING_START      = 0,
    ENUM_ENCODING_G722_1,    = 1
    ENUM_ENCODING_G711_MU,   = 2
    ENUM_ENCODING_G711_A,    = 3
    ENUM_ENCODING_G723,      = 4
    ENUM_ENCODING_MP1L2,     = 5
    ENUM_ENCODING_MP2L2,     = 6
    ENUM_ENCODING_G726,      = 7
    ENUM_ENCODING_AAC,       = 8
    ENUM_ENCODING_RAW        = 100
}NET_EHOME_TALK_ENCODING_TYPE;
```

成员

ENUM_ENCODING_START

编码类型开始

ENUM_ENCODING_G722_1

G722_1

ENUM_ENCODING_G711_MU

G711_MU

ENUM_ENCODING_G711_A

G711_A

ENUM_ENCODING_G723

G723

ENUM_ENCODING_MP1L2

MP1L2

ENUM_ENCODING_MP2L2

MP2L2

ENUM_ENCODING_G726

G726

ENUM_ENCODING_AAC

AAC

ENUM_ENCODING_RAW

RAW

A.2.30 NET_EHOME_TERMINAL_PLAY_STATE

枚举终端的播放状态。

枚举定义

```
typedef enum{
    NET_EHOME_INVAILD_STATE = 0,
    NET_EHOME_HDMI           = 1,
    NET_EHOME_VGA            = 2,
    NET_EHOME_SCHEDULE_PLAY = 3,
    NET_EHOME_SCHEDULE_STOP = 4,
    NET_EHOME_SCREEN_OFF    = 5,
}NET_EHOME_TERMINAL_PLAY_STATE;
```

成员**NET_EHOME_INVAILD_STATE**

无效状态。

NET_EHOME_HDMI

HDMI 状态。

NET_EHOME_VGA

VGA 状态。

NET_EHOME_SCHEDULE_PLAY

开始播放日程。

NET_EHOME_SCHEDULE_STOP

停止播放日程。

NET_EHOME_SCREEN_OFF

关闭屏幕。

A.2.31 NET_EHOME_TERMINAL_TYPE

枚举终端类型。

枚举定义

```
typedef enum{
    NET_EHOME_INVAILD_TERMINAL_TYPE = 0,
    NET_EHOME_ANDROID_STANDARD      = 1,
    NET_EHOME_ANDROID_DECODE        = 2,
    NET_EHOME_ANDROID_TOUCH         = 3,
}NET_EHOME_TERMINAL_TYPE;
```

成员

NET_EHOME_INVAILD_TERMINAL_TYPE

无效类型。

NET_EHOME_ANDROID_STANDARD

安卓标准设备。

NET_EHOME_ANDROID_DECODE

安卓解码设备。

NET_EHOME_ANDROID_TOUCH

安卓触屏设备。

A.2.32 NET_EHOME_THIRD_PARTY_DATA_TYPE

枚举待发送的数据类型。

枚举定义

```
typedef enum{
    NET_EHOME_DATA_INFO_TYPE      = 1,
    NET_EHOME_COMMAND_INFO_TYPE  = 2,
    NET_EHOME_CANCEL_SHOT_PIC    = 3,
    NET_EHOME_START_CAPTURE_PIC  = 4,
    NET_EHOME_STOP_CAPTURE_PIC   = 5,
}NET_EHOME_THIRD_PARTY_DATA_TYPE;
```

成员

NET_EHOME_DATA_INFO_TYPE

叫号数据。

NET_EHOME_COMMAND_INFO_TYPE

叫号控制命令。

NET_EHOME_CANCEL_SHOT_PIC

取消弹图。

NET_EHOME_START_CAPTURE_PIC

开始抓拍。

NET_EHOME_STOP_CAPTURE_PIC

停止显示抓怕的图片。

A.2.33 NET_EHOME_TIME_ZONE

枚举时区。

枚举定义

```
enum NET_EHOME_TIME_ZONE{
    NET_EHOME_ZONE_GMT_LOCAL,
    NET_EHOME_ZONE_GMT_MINUS_12,
    NET_EHOME_ZONE_GMT_MINUS_11,
    NET_EHOME_ZONE_GMT_MINUS_10,
    NET_EHOME_ZONE_GMT_MINUS_9,
    NET_EHOME_ZONE_GMT_MINUS_8,
    NET_EHOME_ZONE_GMT_MINUS_7,
    NET_EHOME_ZONE_GMT_MINUS_6,
    NET_EHOME_ZONE_GMT_MINUS_5,
```

```
NET_EHOME_ZONE_GMT_MINUS_430,  
NET_EHOME_ZONE_GMT_MINUS_4,  
NET_EHOME_ZONE_GMT_MINUS_330,  
NET_EHOME_ZONE_GMT_MINUS_3,  
NET_EHOME_ZONE_GMT_MINUS_2,  
NET_EHOME_ZONE_GMT_MINUS_1,  
NET_EHOME_ZONE_GMT_0,  
NET_EHOME_ZONE_GMT_PLUS_1,  
NET_EHOME_ZONE_GMT_PLUS_2,  
NET_EHOME_ZONE_GMT_PLUS_3,  
NET_EHOME_ZONE_GMT_PLUS_330,  
NET_EHOME_ZONE_GMT_PLUS_4,  
NET_EHOME_ZONE_GMT_PLUS_430,  
NET_EHOME_ZONE_GMT_PLUS_5,  
NET_EHOME_ZONE_GMT_PLUS_530,  
NET_EHOME_ZONE_GMT_PLUS_545,  
NET_EHOME_ZONE_GMT_PLUS_6,  
NET_EHOME_ZONE_GMT_PLUS_630,  
NET_EHOME_ZONE_GMT_PLUS_7,  
NET_EHOME_ZONE_GMT_PLUS_8,  
NET_EHOME_ZONE_GMT_PLUS_9,  
NET_EHOME_ZONE_GMT_PLUS_930,  
NET_EHOME_ZONE_GMT_PLUS_10,  
NET_EHOME_ZONE_GMT_PLUS_11,  
NET_EHOME_ZONE_GMT_PLUS_12,  
NET_EHOME_ZONE_GMT_PLUS_13,  
};
```

成员

NET_EHOME_ZONE_GMT_LOCAL

本地时区。

NET_EHOME_ZONE_GMT_0

UTC 时区。

NET_EHOME_ZONE_GMT_PLUS_8

UTC+8。

A.2.34 NET_EHOME_TRANSMINT_PROTOCOL

枚举传输协议类型。

枚举定义

```
typedef enum{  
    NET_EHOME_TCP      = 1,  
    NET_EHOME_UDP      = 2,  
};
```

```
NET_EHOME_MCAST = 3,  
}NET_EHOME_TRANSMINT_PROTOCOL;
```

成员

NET_EHOME_TCP

TCP

NET_EHOME_UDP

UDP

NET_EHOME_MCAST

MCAST

A.2.35 SEARCH_GET_NEXT_STATUS_ENUM

枚举状态类型。

枚举定义

```
typedef enum tagSEARCH_GET_NEXT_STATUS_ENUM{  
    ENUM_GET_NEXT_STATUS_SUCCESS      = 1000,  
    ENUM_GET_NETX_STATUS_NO_FILE,     = 1001,  
    ENUM_GET_NETX_STATUS_NEED_WAIT,   = 1002,  
    ENUM_GET_NEXT_STATUS_FINISH,      = 1003,  
    ENUM_GET_NEXT_STATUS_FAILED,       = 1004,  
    ENUM_GET_NEXT_STATUS_NOT_SUPPORT = 1005,  
}SEARCH_GET_NEXT_STATUS_ENUM
```

成员

ENUM_GET_NEXT_STATUS_SUCCESS

成功读取到一条数据，处理完本次数据后需要再次调用 FindNext 获取下一条数据。

ENUM_GET_NETX_STATUS_NO_FILE

没有找到一条数据。

ENUM_GET_NETX_STATUS_NEED_WAIT

数据还未就绪，需等待，继续调用 FindNext 函数。

ENUM_GET_NEXT_STATUS_FINISH

数据全部取完。

ENUM_GET_NEXT_STATUS_FAILED

出现异常。

ENUM_GET_NEXT_STATUS_NOT_SUPPORT

设备不支持该操作，不支持的查询类型。

A.2.36 SEARCH_TYPE

枚举查找类型。

枚举定义

```
typedef enum tagSEARCH_TYPE{
    ENUM_SEARCH_TYPE_ERR          = -1,
    ENUM_SEARCH_RECORD_FILE       = 0,
    ENUM_SEARCH_PICTURE_FILE      = 1,
    ENUM_SEARCH_FLOW_INFO         = 2,
    ENUM_SEARCH_DEV_LOG           = 3,
    ENUM_SEARCH_ALARM_HOST_LOG    = 4
}SEARCH_TYPE_ENUM
```

成员

ENUM_SEARCH_TYPE_ERR

错误类型

ENUM_SEARCH_RECORD_FILE

视频文件

ENUM_SEARCH_PICTURE_FILE

图片文件

ENUM_SEARCH_FLOW_INFO

流信息

ENUM_SEARCH_DEV_LOG

日志文件

ENUM_SEARCH_ALARM_HOST_LOG

报警主机的日志文件

附录 B. 附录

B.1 事件类型和详情

表 B-1 通过 ISAPI 上传的事件

事件类型	eventType 取值	pAlarmData (事件报文)
移动侦测	VMD	<u>XML_EventNotificationAlert_MotionDetectionMsg</u>
视频遮挡	shelteralarm	<u>XML_EventNotificationAlert_VideoTamperingAlarmMsg</u>
视频丢失	videoloss	<u>XML_EventNotificationAlert_VideoLossAlarmMsg</u>
报警输入	IO	<u>XML_EventNotificationAlert_AlarmInputAlarmMsg</u>
音频异常侦测	audioexception	<u>XML_EventNotificationAlert_audioException</u>
虚焦侦测	defocus	<u>XML_EventNotificationAlert_DefocusDetectionAlarmMsg</u>
场景变更侦测	scenechangedetection	<u>XML_EventNotificationAlert_sceneChangeDetection</u>
人脸侦测	facetedetection	<u>XML_EventNotificationAlert_FaceDetectionAlarmMsg</u>
入侵侦测	fielddetection	<u>XML_EventNotificationAlert_fielddetection</u>
越界侦测	linedetection	<u>XML_EventNotificationAlert_linedetection</u>
进入区域侦测	regionEntrance	<u>XML_EventNotificationAlert_regionEntrance</u>

事件类型	eventType 取值	pAlarmData（事件报文）
离开区域侦测	regionExiting	<u>XML_EventNotificationAlert_regionExiting</u>
徘徊侦测	loitering	<u>XML_EventNotificationAlert_LoiteringDetectionAlarmMsg</u>
人员聚集侦测	group	<u>XML_EventNotificationAlert_PeopleGatheringDetectionAlarmMsg</u>
快速移动侦测	rapidMove	<u>XML_EventNotificationAlert_FastMovingDetectionMsg</u>
停车侦测	parking	<u>XML_EventNotificationAlert_ParkingDetectionAlarmMsg</u>
物品遗留侦测	unattendedBaggage	<u>XML_EventNotificationAlert_UnattendedBaggageDetectionAlarmMsg</u>
物品拿取侦测	attendedBaggage	<u>XML_EventNotificationAlert_ObjectRemovalDetectionMsg</u>
被动红外报警	PIR	<u>XML_EventNotificationAlert_PIRAlarmMsg</u>
混合目标检测	mixedTargetDetection	<u>JSON_EventNotificationAlert_mixedTargetDetection</u>
人脸抓拍	faceCapture	<u>JSON_EventNotificationAlert_FaceCaptureMsg</u>
排队时长检测	personQueueTime	<u>JSON_EventNotificationAlert_WaitingTimeDetectionMsg</u>
区域排队人数检测	personQueueCounting	<u>JSON_EventNotificationAlert_PeopleQueueingUpAlarmMsg</u>
人脸图片比对结果	alarmResult	<u>JSON_EventNotificationAlert_FacePicComparisonMsg</u>

事件类型	eventType 取值	pAlarmData（事件报文）
人脸建模结果	faceSnapModeling	<u>JSON_EventNotificationAlert_FaceModelingMsg</u>
车辆检测（自动车牌识别）	ANPR	<u>XML_EventNotificationAlert_ANPRMsg</u>
非法摆摊检测	cityManagement	<u>JSON_EventNotificationAlert_UnregisteredStreetVendorMsg</u>
客流统计报告	PeopleCounting	<u>XML_EventNotificationAlert_PeopleCountingEventMsg</u>
热度图报告	heatmap	<u>JSON_EventNotificationAlert_HeatMapMsg</u>
温度报警	TMA	<u>XML_EventNotificationAlert_TemperatureAlarmMsg</u>
硬盘错误报警	diskerror	<u>XML_EventNotificationAlert_HDDErrorAlarmMsg</u>
非法登录报警	illaccess	<u>XML_EventNotificationAlert_IllegalLoginAlarmMsg</u>
电池状态事件	batteryStatus	<u>JSON_EventNotificationAlert_BatteryStatus</u>

B.2 ISUPSDK 错误码

以下错误码可能会在集成过程中返回。可以根据返回的错误码搜索错误类型和描述从而解决问题。

通用错误码

错误名	错误码	描述
NET_DVR_NOERROR	0	无错误。
NET_DVR_PASSWORD_ERROR	1	用户名或密码不正确。

错误名	错误码	描述
NET_DVR_NOENOUGHPRI	2	没有权限。
NET_DVR_NOINIT	3	SDK 未初始化。
NET_DVR_CHANNEL_ERROR	4	通道号不正确。
NET_DVR_OVER_MAXLINK	5	无法再添加设备。
NET_DVR_VERSIONNOMATCH	6	SDK 和设备版本不匹配。
NET_DVR_NETWORK_FAIL_CONNECT	7	连接到设备失败。设备掉线或网络连接断开。
NET_DVR_NETWORK_SEND_ERROR	8	发送数据失败。
NET_DVR_NETWORK_RECV_ERROR	9	从设备接收数据失败。
NET_DVR_NETWORK_RECV_TIMEOUT	10	从设备接收数据超时。
NET_DVR_NETWORK_ERRORDATA	11	数据错误。
NET_DVR_ORDER_ERROR	12	调用顺序错误。
NET_DVR_OPERNOPERMIT	13	没有权限。
NET_DVR_COMMANDTIMEOUT	14	设备执行命令超时。
NET_DVR_PARAMETER_ERROR	17	参数无效。
NET_DVR_NOSUPPORT	23	设备不支持。
NET_DVR_DVROPRATEFAILED	29	操作失败。
NET_DVR_DIR_ERROR	40	路径不正确。
NET_DVR_ALLOC_RESOURCE_ERROR	41	SDK 资源分配错误。
NET_DVR_AUDIO_MODE_ERROR	42	声卡模式不匹配。
NET_DVR_NOENOUGH_BUF	43	数据或图片缓冲区不足。
NET_DVR_CREATESOCKET_ERROR	44	创建套接字失败。
NET_DVR_SETSOCKET_ERROR	45	设置套接字失败。

错误名	错误码	描述
NET_DVR_MAX_NUM	46	没有可连接的设备。
NET_DVR_USERNOTEXIST	47	该用户不存在。
NET_DVR_GETLOCALIPANDMACFAIL	53	获取本地 IP 或 Mac 地址失败。
NET_DVR_VOICEMONOPOLIZE	69	声卡已被占用。
NET_DVR_CREATEDIR_ERROR	71	创建日志文件目录失败。
NET_DVR_BINDSOCKET_ERROR	72	绑定套接字失败。
NET_DVR_SOCKETCLOSE_ERROR	73	套接字掉线。
NET_DVR_USERID_ISUSING	74	注销失败。该用户正在进行其他操作。
NET_DVR_SOCKETLISTEN_ERROR	75	监听失败。
NET_DVR_CONVERT_SDK_ERROR	85	加在动态转码链接库 (systemTransform.dll) 失败。
NET_DVR_FUNCTION_NOT_SUPPORT_OS	98	该操作系统不支持该功能。
NET_DVR_USE_LOG_SWITCH_FILE	103	日志已启用。
NET_DVR_PACKET_TYPE_NOT_SUPPORT	105	码流封装格式无效。
NET_DVR_IPPARA_IPID_ERROR	106	网络接入配置时 IP 地址有误。
NET_DVR_STREAM_ENCRYPT_CHECK_FAIL	130	码流加密校验失败
NET_DVR_STREAM_STATUS_NOT_INIT	131	支持 5.0 版本 ISUP 的设备未上传码流初始信息。
NET_DVR_CERTIFICATE_FILE_ERROR	147	证书错误。
NET_DVR_LOAD_SSL_LIB_ERROR	148	加载 SSL 库失败。

错误名	错误码	描述
NET_DVR_LOAD_ANALYZE_DATA_LIB_ERROR	149	加载帧分析库失败。
NET_DVR_LOAD_LIBEAY32_DLL_ERROR	156	加载 libeay32.dll 库失败。
NET_DVR_LOAD_SSLEAY32_DLL_ERROR	157	加载 ssleay32.dll 库失败。
NET_ERR_LOAD_LIBICONV	158	加载 libiconv2.dll 库失败。
NET_ERR_SSL_CONNECT_FAILED	159	SSL 连接失败。
NET_ERR_LOAD_ZLIB	161	加载 zlib.dll 库失败。
NET_PREVIEW_ERR_CHANNEL_BUSY	165	该通道已在发流。
NET_PREVIEW_ERR_CLIENT_BYSY	166	取流地址重复。
NET_PREVIEW_ERR_STREAM_UNSupport	167	不支持的码流类型。
NET_PREVIEW_ERR_TRANSPORT_UNSupport	168	不支持的传输方式。
NET_PREVIEW_ERR_CONNECT_SERVER_FAIL	169	连接预览流媒体服务器失败。
NET_PREVIEW_ERR_QUERY_WLAN_INFO_FAIL	170	查询设备公网出口地址失败。
NET_PREVIEW_ERR_NO_VIDEO_FAIL	171	无视频源。
NET_PREVIEW_ERR_SET_ENCODE_PARAM_FAIL	172	设置编码参数失败。
NET_PREVIEW_ERR_SET_PACK_TYPE_FAIL	173	设置码流分支类型失败。

错误名	错误码	描述
NET_PREVIEW_ERR_NOW_IN_PREVIEW_FAIL	174	设备已经在取流，不再支持预取流。
NET_PREVIEW_ERR_NOW_IN_PRESTREAM_FAIL	175	设备已经在预取流，不再支持预取流。
NET_PREVIEW_ERR_BREAKOFF_PRESTREAM_FAIL	176	设备触发另一路取流，断开前一路的预取流。
NET_PREVIEW_ERR_P2P_NOT_FOUND	177	P2P 取流通道不存在。
NET_PLAYBACK_ERR_TYPE_UNSUPPORTED	3501	该回放类型不支持。
NET_PLAYBACK_ERR_NO_FILE_MATCHED	3502	未找到文件。
NET_PLAYBACK_ERR_START_TIME_ERROR	3503	开始时间错误。
NET_PLAYBACK_ERR_STOP_TIME_ERROR	3504	结束时间错误。
NET_PLAYBACK_ERR_NO_FILE_FOUND	3505	该时间段内无录像。
NET_PLAYBACK_ERR_CONNECT_SERVER_FAIL	3506	连接至回放服务失败。
NET_SS_CLIENT_ERR_KMS_TOKEN_FAIL	3601	获取 token 失败。
NET_SS_CLIENT_ERR_KMS_UPLOAD_FAIL	3602	上传图片至 KMS 失败。
NET_SS_CLIENT_ERR_CLOUD_POOLIST_FAIL	3603	获取资源池失败。

错误名	错误码	描述
NET_SS_CLIENT_ERR_CLOUD_BESTNODE_FAIL	3604	获取最优节点失败。
NET_SS_CLIENT_ERR_DELETE_PICTURE_FAIL	3606	删除文件失败。
NET_SS_CLIENT_ERR_PROTO_UNSAFE	3607	安全模式下不支持非安全协议。
NET_SS_CLIENT_ERR_FILE_INEXISTED	3608	待删除或下载的图片不存在。
NET_SS_CLIENT_ERR_AUTH_FAILED	3609	认证失败。
NET_SS_CLIENT_ERR_UPLOAD_FAIL	3610	存储组件的客户端 Cloud\VRB\Tomcat 图片上传失败
NET_DVR_ERR_GENERAL_UNKNOW_ERROR	3701	未知错误。
NET_DVR_ERR_GENERAL_PARSE_FAILED	3702	报文解析错误。
NET_DVR_ERR_GENERAL_SYSTEM_ERROR	3703	系统内部错误(比如设备系统调用出错)。
NET_DVR_ERR_GENERAL_COMMAND_UNKNOW	3704	非法命令。
NET_DVR_ERR_GENERAL_COMMAND_NO_LONGER_SUPPORTED	3705	过时命令。
NET_DVR_ERR_GENERAL_COMMAND_NOT_SUITABLE	3706	错误命令。
NET_DVR_ERR_GENERAL_COMMAND_NOT_ALLOW	3707	未授权命令。
NET_DVR_ERR_GENERAL_CHECKSUM_ERROR	3708	校验码错误。

错误名	错误码	描述
NET_DVR_ERR_GENERAL_HEADER_INVALID	3709	消息头非法。
NET_DVR_ERR_GENERAL_LENGTH_INVALID	3710	消息长度非法。
NET_DVR_ERR_GENERAL_PU_BUSY	3711	设备无法响应。
NET_DVR_ERR_GENERAL_OPERATION_FAILED	3712	操作码错误。
NET_DVR_ERR_GENERAL_PU_NO_CRYPTO_FOUND	3713	设备或平台未找到对应的加密算法。
NET_DVR_ERR_GENERAL_PU_NO_RESOURCE	3715	没有可用资源。
NET_DVR_ERR_GENERAL_PU_CHANNEL_ERROR	3716	通道错误。
NET_DVR_ERR_GENERAL_SYSTEM_COMMAND_PU_COMMAND_UNSUPPORTED	3717	不支持的命令。
NET_DVR_ERR_GENERAL_SYSTEM_COMMAND_PU_NO_RIGHTS_TO_DO_COMMAND	3718	没有权限。
NET_DVR_ERR_GENERAL_NO_SESSION_FOUND	3719	没有找到会话。
NET_DVR_ERR_GENERAL_PU_NO_VALID_PRELINK	3720	没有可用的 P2P 预链接资源。
NET_DVR_ERR_GENERAL_PU_NO_INNER_RESOURCE	3721	没有可用的直连资源。

错误名	错误码	描述
NET_DVR_ERR_GENERAL_PU_NO_P2P_RESOURCE	3722	没有可用的 P2P 资源。
NET_DVR_ERR_GENERAL_PU_NO_UESR	3723	设备未绑定用户。
NET_DVR_ERR_GENERAL_TICKET_EXPIRED	3724	访问凭证过期。
NET_DVR_ERR_GENERAL_TICKET_INVALID	3725	访问凭证无效。
NET_DVR_ERR_GENERAL_NO_P2PSERVER_RESOURCE	3726	无可用的 P2P 服务。
NET_DVR_ERR_GENERAL_PU_NOT_FOUND	3727	未找到设备或设备在 P2P 服务器上已下线。
NET_DVR_ERR_GENERAL_SESSION_FREED	3728	会话已释放。
NET_DVR_ERR_RECORD_SEARCH_START_TIME_ERROR	3729	查找录像开始时间错误。
NET_DVR_ERR_RECORD_SEARCH_STOP_TIME_ERROR	3730	查找录像结束时间错误。
NET_DVR_ERR_RECORD_SEARCH_FAIL	3731	查找录像失败。
NET_DVR_ERR_RECORD_NO_RESOURCE	3732	录像失败，无可用的存储资源。
NET_DVR_ERR_CAPTURE_PIC_LOCAL_FAILED	3733	设备本地抓图失败。
NET_DVR_ERR_CAPTURE_PIC_APPLY_CACHE_FAILED	3734	图片缓存申请失败。

错误名	错误码	描述
NET_DVR_ERR_CAPTURE_PIC_PARSE_PMS_DOMAIN_FAILED	3735	PMS 域名解析错误。
NET_DVR_ERR_CAPTURE_PIC_CONNECT_PMS_FAILED	3736	PMS 连接失败。
NET_DVR_ERR_CAPTURE_PIC_CREATE_PMS_PACKET_FAILED	3737	创建 PMS 报文错误。
NET_DVR_ERR_CAPTURE_PIC_SEND_PMS_FAILED	3738	PMS 发送数据错误。
NET_DVR_ERR_CAPTURE_PIC_RECV_PMS_FAILED	3739	PMS 接收数据错误。
NET_DVR_ERR_CAPTURE_PIC_PARSE_PMS_RESPONSE_FAILED	3740	PMS 应答报文解析错误。
NET_DVR_ERR_CAPTURE_PIC_GET_URL_FAILED	3741	获取 URL 失败。
NET_ERR_DEV_SLEEP	8301	设备当前处于休眠状态  说明 当前状态下设备无法对预览、回放、语音对讲、配置等 CMS 中的信令作响应。
NET_ERR_DEV_GOINGTOSLEEP	8302	即将进入休眠状态，预计 13 秒  说明 此时无法唤醒设备，直至执行休眠动作结束，设备正式进入休眠状态，才可唤醒。

错误名	错误码	描述
NET_DVR_INIT_CONFIG_ERROR	100000	初始化配置接口未调用。
NET_DVR_LOAD_COMPONENT_ERROR	100001	组件库加载失败。

语音对讲错误码

错误名	错误码	描述
NET_AUDIOINTERCOM_OK	600	无错误。
NET_AUDIOINTECOM_ERR_NOTSUPPORT	601	不支持。
NET_AUDIOINTECOM_ERR_ALLOC_MEMORY	602	申请内存失败。
NET_AUDIOINTECOM_ERR_PARAMETER	603	参数值不正确。
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	调用顺序不正确。
NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	未找到设备。
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	设备打开失败。
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	上下文不正确。
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV 文件错误。
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	WAV 数据类型无效。

错误名	错误码	描述
NET_AUDIOINTECOM_ERR_ENCODE_FAIL	610	编码失败。
NET_AUDIOINTECOM_ERR_DECODE_FAIL	611	解码失败。
NET_AUDIOINTECOM_ERR_NO_PLAYBACK	612	语音播放失败。
NET_AUDIOINTECOM_ERR_DENOISE_FAIL	613	降噪失败。
NET_AUDIOINTECOM_ERR_UNKOWN	619	未知错误。

B.3 ISAPI 返回码

ISAPI 集成过程中的返回码是根据 HTTP 协议的状态码进行分类的。预先规定的状态码包括 1-正常, 2-设备繁忙, 3-设备错误, 4-操作无效, 5-报文格式无效, 6-报文内容无效, 7-需要重启设备, 8-批量操作, 9-其他错误 (暂无子状态码)。此外每个状态码包含多个子状态码, 返回码和子状态码存在一一对应关系。

- statusCode=1 (正常)
- statusCode=2 (设备繁忙)
- statusCode=3 (设备错误)
- statusCode=4 (操作无效)
- statusCode=5 (报文格式无效)
- statusCode=6 (报文内容无效)
- statusCode=7 (重启)
- statusCode=8 (批量操作)

statusCode=1

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
ok	0x1	操作完成。
riskPassword	0x10000002	风险密码。
armProcess	0x10000005	布防进度。

statusCode=2

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
noMemory	0x20000001	设备内存不足。
serviceUnavailable	0x20000002	服务不可用。
upgrading	0x20000003	升级中。
deviceBusy	0x20000004	设备正忙或设备无响应。
reConnectIpc	0x20000005	重新连接视频服务器。
transferUpgradePackageFailed	0x20000006	设备升级数据传输操作失败。
startUpgradeFailed	0x20000007	设备启动升级失败。
getUpgradeProcessfailed.	0x20000008	获取升级状态失败。
certificateExist	0x2000000B	身份验证证书已存在。

statusCode=3

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
deviceError	0x30000001	硬件错误。
badFlash	0x30000002	操作 Flash 错误。
28181Uninitialized	0x30000003	未初始化 28181 配置。
socketConnectError	0x30000005	套接字连接失败。
receiveError	0x30000007	接收响应报文失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
deletePictureError	0x3000000A	删除图片失败。
pictureSizeExceedLimit	0x3000000C	图片大小超出限制。
clearCacheError	0x3000000D	清除缓存失败。
updateDatabasError	0x3000000F	更新数据库失败。
searchDatabaseError	0x30000010	数据库查找失败。
writeDatabaseError	0x30000011	数据库写入失败。
deleteDatabaseError	0x30000012	数据库元素删除失败。
searchDatabaseElementError	0x30000013	获取数据库元素个数失败。
cloudAutoUpgradeException	0x30000016	自动云端下载升级包升级失败。
HBPException	0x30001000	HBP 异常。
UDEPException	0x30001001	UDEP 异常。
elasticSearchException	0x30001002	Elastic 异常。
kafkaException	0x30001003	Kafka 异常。
HBaseException	0x30001004	Hbase 异常。
sparkException	0x30001005	Spark 异常。
yarnException	0x30001006	Yarn 异常。
cacheException	0x30001007	Cache 异常。
trafficException	0x30001008	卡口大数据服务器异常。
faceException	0x30001009	人脸大数据服务器异常。
wifiException	0x3000100A	Wi-Fi 大数据服务器异常。
structException	0x3000100D	视频参数结构服务器异常。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
SSDFileSystemsError	0x30001013	SSD 文件系统错误 (非 EXT4 文件系统时报错)。
insufficientSSDCapacityForFPD	0x30001014	用于人员频次检测的 SSD 空间不足。
noLinkageResource	0x30001015	联动资源不足。
noArmingResource	0x30001016	布防资源不足。
engineAbnormal	0x30002015	引擎异常
engineInitialization	0x30002016	引擎初始化中
algorithmLoadingFailed	0x30002017	模型加载失败
algorithmDownloadFailed	0x30002018	模型下载失败
algorithmDecryptionFailed	0x30002019	模型解密失败
unboundChannel	0x30002020	加载新模型需要删除通道绑定
unsupportedResolution	0x30002021	不支持的分辨率
unsupportedStreamType	0x30002022	不支持的码流类型
insufficientDecRes	0x30002023	解码资源不足
insufficientEnginePerformance	0x30002024	引擎性能不足 (超过可分析通道数)
improperResolution	0x30002025	图片超过分辨率 (最大 4096*4096)
improperPicSize	0x30002026	图片太大 (最大支持 5M)
URLDownloadFailed	0x30002027	图片下载失败 (URL 下载)
unsupportedImageFormat	0x30002028	图片格式不支持 (目前只支持 jpg)

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
unsupportedPollingIntervalTime	0x30002029	轮询间隔太低 (需大于 10s)
exceedImagesNumber	0x30002030	超出图片数量限制 (平台每次下发图片数量 1-100 张图片 URL, 最大 100)
unsupportedMPID	0x30002031	下发的 MPID, 设备中不存在, 不支持针对这个 MPID 的更新操作
modelPackageNotMatchLabel	0x30002032	模型与描述文件不匹配
modelPackageNotMatchTask	0x30002033	任务和模型种类不匹配
insufficientSpace	0x30002034	剩余空间不足。(当数量未达上限但模型包大小超过剩余存储空间大小时, 无法添加)
engineUnLoadingModelPackage	0x30002035	任务下发失败。该引擎未绑定模型。(解绑失败, 当前引擎没有绑定模型。)
engineWithModelPackage	0x30002036	绑定失败, 该引擎已绑定模型, 请先解绑
modelPackageDelete	0x30002037	绑定失败, 该模型已被删除。
deleteTaskFailed	0x30002038	清除任务失败 (用户关闭任务失败时返回)
modelPackageNumberslimited	0x30002039	模型包添加失败, 模型包个数已达到上限。
modelPackageDeleteFailed	0x30002040	模型包删除失败。
calibrationTimeout	0x30002051	标定超时。
captureTimeout	0x30006000	数据采集超时。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
lowScore	0x30006001	采集的数据质量低。
uploadingFailed	0x30007004	上传失败。

statusCode=4

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
notSupport	0x40000001	不支持。
lowPrivilege	0x40000002	没有权限。
badAuthorization	0x40000003	身份验证失败。
methodNotAllowed	0x40000004	操作方法无效。
notSetHdiskRedund	0x40000005	无法设置硬盘磁盘冗余属性。
invalidOperation	0x40000006	操作无效。
notActivated	0x40000007	设备未激活。
hasActivated	0x40000008	设备已激活。
certificateAlreadyExist	0x40000009	证书已存在。
USBNotExist	0x40000010	USB 设备未连接。
operateFailed	0x4000000F	操作失败。
upgradePackageMorethan2GB	0x40001000	请上传小于 2G 的升级包。
IDNotExist	0x40001001	ID 不存在。
interfaceOperationError	0x40001002	接口操作失败。
synchronizationError	0x40001003	同步失败。
synchronizing	0x40001004	正在同步。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
importError	0x40001005	导入失败。
importing	0x40001006	正在导入。
fileAlreadyExists	0x40001007	该文件已存在。
invalidID	0x40001008	ID 无效。
backupnodeNotAllowe Log	0x40001009	备节点不允许登录。
exportingError	0x4000100A	导出失败。
exporting	0x4000100B	正在导出。
exportEnded	0x4000100C	导出终止。
exported	0x4000100D	已导出。
IPOccupied	0x4000100E	IP 地址已被占用。
IDAlreadyExists	0x4000100F	该 ID 已存在。
exportItemsExceedLimi t	0x40001010	导出个数已超上限。
noFiles	0x40001011	文件不存在。
beingExportedByAnoth erUser	0x40001012	其他人员正在导出。
needReAuthentication	0x40001013	升级后需要进行身份验证。
unitAddNotOnline	0x40001015	添加的数据分析服务器不在线。
unitControl	0x40001016	数据分析服务器已被添加。
analysis unitFull	0x40001017	数据分析服务器添加已满。
unitIDError	0x40001018	数据分析服务器 ID 不存在。
unitExit	0x40001019	数据分析服务器已经存在于列表中。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
unitSearch	0x4000101A	数据分析服务器列表查询失败。
unitNotOnline	0x4000101B	数据分析服务器不在线。
unitInfoError	0x4000101C	获取数据分析服务器信息失败。
unitGetNodeInfoError	0x4000101D	获取节点信息失败。
unitGetNetworkInfoError	0x4000101E	获取数据分析服务器的网络信息失败。
unitSetNetworkInfoError	0x4000101F	设置数据分析服务器的网络信息失败。
setSmartNodeInfoError	0x40001020	设置节点信息失败。
setUnitNetworkInfoError	0x40001021	设置数据分析服务器网络信息失败。
unitRestartCloseError	0x40001022	重启或关闭数据分析服务器失败。
virtualIPnotAllowed	0x40001023	不允许添加虚拟 IP 地址。
unitInstalled	0x40001024	已安装数据分析服务器。
badSubnetMask	0x40001025	子网掩码无效。
uintVersionMismatched	0x40001026	数据分析服务器版本不匹配。
deviceModelMismatched	0x40001027	添加失败。设备型号不匹配。
unitAddNotSelf	0x40001028	不允许添加外围设备。
noValidUnit	0x40001029	不存在有效的数据分析服务器。
unitNameDuplicate	0x4000102A	数据分析服务器名称重复。
deleteUnitFirst	0x4000102B	请先删除该节点上已添加的数据分析服务器。
getLocalInfoFailed	0x4000102C	获取服务器信息失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
getClientAddedNodeFailed	0x4000102D	获取数据分析服务器已添加节点信息失败。
taskExit	0x4000102E	任务已经存在。
taskInitError	0x4000102F	任务初始化失败。
taskSubmitError	0x40001030	任务提交失败。
taskDelError	0x40001031	任务删除失败。
taskPauseError	0x40001032	任务暂停失败。
taskContinueError	0x40001033	任务启动失败。
taskSeverNoCfg	0x40001035	全文检索服务器未配置。
taskPicSeverNoCfg	0x40001036	图片服务器未配置。
taskStreamError	0x40001037	取流信息异常。
taskRecSDK	0x40001038	不支持历史录像。
taskCasaError	0x4000103A	不支持级联。
taskVCARuleError	0x4000103B	智能规则无效。
taskNoRun	0x4000103C	任务未执行。
unitLinksNoStorageNode	0x4000103D	数据分析服务器未关联节点。请先配置节点。
searchFailed	0x4000103E	录像文件搜索失败。
searchNull	0x4000103F	没有录像片段。
userScheOffline	0x40001040	任务调度服务不在线。
updateTypeUnmatched	0x40001041	升级包类型不匹配。
userExist	0x40001043	用户已存在。
userCannotDelAdmin	0x40001044	不能删除管理员用户。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
userInexistence	0x40001045	用户名不存在。
userCannotCreatAdmin	0x40001046	不能创建管理员用户。
monitorCamExceed	0x40001048	最多布控 3000 个监控点。
monitorCunitOverLimit	0x40001049	添加失败。控制中心最多嵌套 5 层。
monitorReginOverLimit	0x4000104A	添加失败。区域最多嵌套 5 层。
monitorArming	0x4000104B	该监控点已关联布控。请先解除布控, 然后重试。
monitorSyncCfgNotSet	0x4000104C	管理平台参数未配置。
monitorFdSyncing	0x4000104E	正在同步。请等同步完成后再进行操作。
monitorParseFailed	0x4000104F	解析监控点信息失败。
monitorCreatRootFailed	0x40001050	创建资源节点失败。
deleteArmingInfo	0x40001051	该监控点已布控。请先接触布控再操作。
cannotModify	0x40001052	不允许修改。请重新选择。
cannotDel	0x40001053	不允许删除。请重新选择。
deviceExist	0x40001054	设备已存在。
IPErrorConnectFailed	0x40001056	连接失败。请检查网络端口。
cannotAdd	0x40001057	只支持添加抓拍机。
serverExist	0x40001058	服务器已存在。
fullTextParamError	0x40001059	全文检索参数错误。
storParamError	0x4000105A	存储服务器参数错误。
picServerFull	0x4000105B	图片服务器存储空间已满。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
NTPUnconnect	0x4000105C	NTP 服务器连接失败。请检查参数。
storSerConnectFailed	0x4000105D	存储服务器连接失败。请检查网络端口。
storSerLoginFailed	0x4000105E	存储服务器登录失败。请检查用户名和密码。
searchSerConnectFailed	0x4000105F	全文检索服务器连接失败。请检查网络端口。
searchSerLoginFailed	0x40001060	全文检索服务器登录失败。请检查用户名和密码。
kafkaConnectFailed	0x40001061	Kafka 连接失败。请检查网络端口。
mgmtConnectFailed	0x40001062	管理平台连接失败。请检查网络端口。
mgmtLoginFailed	0x40001063	管理平台登录失败。请检查用户名和密码。
TDAConnectFailed	0x40001064	交通数据接入服务器连接失败。请检查服务器状态。
86sdkConnectFailed	0x40001065	8600 平台的监听端口连接失败。请检查参数配置。
nameExist	0x40001066	存在同名服务器。
batchProcessFailed	0x40001067	批量处理失败。
IDNotExist	0x40001068	服务器 ID 不存在。
serviceNumberReachesLimit	0x40001069	服务添加已达上限。
invalidServiceType.	0x4000106A	服务类型错误。
clusterGetInfo	0x4000106B	获取集群分组信息失败。
clusterDelNode	0x4000106C	删除节点失败。
clusterAddNode	0x4000106D	添加节点失败。
clusterInstalling	0x4000106E	集群组件中……请勿操作。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
clusterUninstall	0x4000106F	集群重置中……请勿操作。
clusterInstall	0x40001070	组建集群失败。
clusterIpError	0x40001071	任务调度服务器的 IP 地址无效。
clusterNotSameSeg	0x40001072	主备节点需要在同一网段。
clusterVirIpError	0x40001073	自动获取虚拟 IP 地址失败。请手动输入。
clusterNodeUnadd	0x40001074	指定的主（备）节点未被添加。
clusterNodeOffline	0x40001075	任务调度服务器不在线。
nodeNotCurrentIP	0x40001076	添加主备节点时，需要选择当前页面 IP 对应的分析节点。
addNodeNetFailed	0x40001077	添加节点失败。网络连接失败。
needTwoMgmtNode	0x40001078	组建主备节点时，需要两个管理节点。
ipConflict	0x40001079	虚拟 IP 和数据分析服务器的 IP 冲突。
ipUsed	0x4000107A	虚拟 IP 地址已被占用。
cloudAnalyseOnline	0x4000107B	云分析服务器在线。
virIP&mainIPnotSame NetSegment	0x4000107C	虚拟 IP 地址和主备节点 IP 地址不在同一网段。
getNodeDispatchInfoFailed	0x4000107D	获取节点调度信息失败。
unableModifyManagementNetworkIP	0x4000107E	修改管理网口失败。分析板在集群中。
notSpecifyVirtualIP	0x4000107F	主备集群须指定虚拟 IP 地址。
armingFull	0x40001080	布控设备已达上限。
armingNoFind	0x40001081	布控信息不存在。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
disArming	0x40001082	解除布控失败。
getArmingError	0x40001084	获取布控信息失败。
refreshArmingError	0x40001085	刷新布控信息失败。
ArmingPlateSame	0x40001086	布控的车牌号码相同。
ArmingParseXLSError	0x40001087	解析布控信息文件失败。
ArmingTimeError	0x40001088	布控时间段错误。
ArmingSearchTimeError	0x40001089	检索时间段错误。
armingRelationshipReachesLimit	0x4000108A	布控关系个数已达上限。
duplicateArmingName	0x4000108B	布控关系名已存在。
noMoreArmingListAdded	0x4000108C	布控黑名单库已达上限。
noMoreCamerasAdded	0x4000108D	布控监控点已达上限。
noMoreArmingListAddedWithCamera	0x4000108E	监控点关联库已达上限。
noMoreArmingPeriodAdded	0x4000108F	布控时间段已达上限。
armingPeriodsOverlapped	0x40001090	布控时间段重叠。
noArmingAlarmInfo	0x40001091	布控报警信息不存在。
armingAlarmUnRead	0x40001092	获取未读布控报警信息个数失败。
getArmingAlarmError	0x40001093	获取布控报警信息失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
searchByPictureTimed Out	0x40001094	以图搜图任务超时。请重新检索。
comparisonTimeRange Error	0x40001095	比对时间范围错误。
selectMonitorNumber UpperLimit	0x40001096	卡口 ID 过滤超过上限。
noMoreComparisonTas ksAdded	0x40001097	同时比对任务超过上限。
GetComparisonResultF ailed	0x40001098	获取比对结果失败。
comparisonTypeError	0x40001099	比对类型错误。
comparisonUnfinished	0x4000109A	比对未完成。
facePictureModelInvali d	0x4000109B	人脸模型无效。
duplicateLibraryName.	0x4000109C	库名已存在。
noRecord	0x4000109D	无该条记录。
countingRecordsFailed.	0x4000109E	记录数统计失败。
getHumanFaceFrameF ailed	0x4000109F	根据图片获取人脸框失败。
modelingFailed.	0x400010A0	根据人脸图片 URL 建模失败。
1V1FacePictureCompar isonFailed	0x400010A1	1V1 人脸相似度比对失败。
libraryArmed	0x400010A2	黑名单库已布控。
licenseExeedLimit	0x400010A3	加密狗限制。
licenseExpired	0x400010A4	加密狗过期

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
licenseDisabled	0x400010A5	加密狗不可用。
licenseNotExist	0x400010A6	加密狗不存在。
SessionExpired	0x400010A7	会话过期。
beyondConcurrentLimit	0x400010A8	超出并发限制。
stopSync	0x400010A9	停止同步。
getProgressFailed	0x400010AA	获取进度失败。
uploadExtraCaps	0x400010AB	上传文件超上限。
timeRangeError	0x400010AC	时间范围错误。
dataPortNotConnected	0x400010AD	数据口未连接。
addClusterNodeFailed	0x400010AE	添加到集群失败。该设备已被其他集群添加。
taskNotExist	0x400010AF	任务不存在。
taskQueryFailed	0x400010B0	任务查询失败
modifyTimeRuleFailed	0x400010B2	任务已存在。不允许修改时间规则。
modifySmartRuleFailed	0x400010B3	任务已经存在。不允许修改 VAC 规则。
queryHistoryVideoFailed	0x400010B4	查询历史录像失败。
addDeviceFailed	0x400010B5	添加设备失败。
addVideoFailed	0x400010B6	添加录像文件失败。
deleteAllVideoFailed	0x400010B7	删除所有录像文件失败。
createVideoIndexFailed	0x400010B8	录像文件创建检索失败。
videoCheckTypeFailed	0x400010B9	录像文件类型校验失败。
configStructuredAddressesFailed	0x400010BA	配置结构化服务器的 IP 地址失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
configPictureServerAddressFailed	0x400010BB	配置图片存储服务器的 IP 地址失败。
storageServiceIPNotExist	0x400010BD	存储服务器 IP 地址不存在。
syncBackupDatabaseFailed	0x400010BE	同步备机数据库失败。请重试。
syncBackupNTPTimeFailed	0x400010BF	同步备机 NTP 时间失败。
clusterNotSelectLoopbackAddress	0x400010C0	主备集群不支持环回地址。
addFaceRecordFailed	0x400010C1	人脸记录添加失败。
deleteFaceRecordFailed	0x400010C2	人脸记录删除失败。
modifyFaceRecordFailed	0x400010C3	人脸记录修改失败。
queryFaceRecordFailed	0x400010C4	人脸记录查询失败。
faceDetectFailed	0x400010C5	人脸检测失败。
libraryNotExist	0x400010C6	库不存在。
blackListQueryExporting	0x400010C7	正在导出匹配的黑名单。
blackListQueryExported	0x400010C8	匹配的黑名单已经导出。
blackListQueryStopExporting	0x400010C9	停止导出匹配的黑名单。
blackListAlarmQueryExporting	0x400010CA	正在导出匹配的黑名单警报。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
blackListAlarmQueryExported	0x400010CB	匹配的黑名单警报已经导出。
blackListAlarmQueryStopExporting	0x400010CC	停止导出匹配的黑名单警报。
getBigDataCloudAnalysisFailed	0x400010CD	获取大数据云分析失败。
setBigDataCloudAnalysisFailed	0x400010CE	配置大数据云分析失败。
submitMapSearchFailed	0x400010CF	提交以图搜图任务失败。
controlRelationshipNotExist	0x400010D0	布控关系不存在。
getHistoryAlarmInfoFailed	0x400010D1	获取历史报警信息失败。
getFlowReportFailed	0x400010D2	获取客流统计报告失败。
addGuardFailed	0x400010D3	布控配置添加失败。
deleteGuardFailed	0x400010D4	布控配置删除失败。
modifyGuardFailed	0x400010D5	布控配置修改失败。
queryGuardFailed	0x400010D6	布控配置查询失败。
uploadUserSuperCaps	0x400010D7	用户信息上传已达上限。
bigDataServerConnectFailed	0x400010D8	大数据连接失败。
microVideoCloudRequestInfoBuildFailed	0x400010D9	微视云请求信息组建失败。
microVideoCloudResponseInfoBuildFailed	0x400010DA	微视云响应信息解析失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
transcodingServerRequestInfoBuildFailed	0x400010DB	转码服务器请求信息组建失败。
transcodingServerResponseInfoParseFailed	0x400010DC	转码服务器响应信息解析失败。
transcodingServerOffline	0x400010DD	转码服务器不在线。
microVideoCloudOffline	0x400010DE	微视云服务不在线。
UPSServerOffline	0x400010DF	UPS 检测服务不在线。
statisticReportRequestInfoBuildFailed	0x400010E0	统计报告请求信息组建失败。
statisticReportResponseInfoParseFailed	0x400010E1	统计报告响应信息解析失败。
DisplayConfigInfoBuildFailed	0x400010E2	显示配置信息组建失败。
DisplayConfigInfoParseFailed	0x400010E3	显示配置信息解析失败。
DisplayConfigInfoSaveFailed	0x400010E4	显示配置信息保存失败。
notSupportDisplayConfigType	0x400010E5	不支持该显示配置类型。
passError	0x400010E7	密码错误。
upgradePackageLarge	0x400010EB	升级包过大。
sesssionUserReachesLimit	0x400010EC	通过会话登录用户数达到上限。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
ISO 8601TimeFormatError	0x400010ED	ISO8601 时间格式错误。
clusterDissolutionFailed	0x400010EE	集群解散失败。
getServiceNodeInfoFailed	0x400010EF	服务节点信息获取失败。
getUPSInfoFailed	0x400010F0	UPS 配置信息获取失败。
getDataStatisticsReportFailed	0x400010F1	数据统计报告获取失败。
getDisplayConfigInfoFailed	0x400010F2	显示配置获取失败。
namingAnalysisBoardNotAllowed	0x400010F3	不允许修改分析板名称。
onlyDrawRegionsOfConvexPolygon	0x400010F4	仅支持绘制凸多边形区域。
bigDataServerResponseInfoParseFailed	0x400010F5	大数据服务的响应信息解析失败。
bigDataServerReturnFailed	0x400010F6	大数据服务返回失败。
microVideoReturnFailed	0x400010F7	微视云服务返回失败。
transcodingServerReturnFailed	0x400010F8	转码服务返回失败。
UPSServerReturnFailed	0x400010F9	UPS 检测服务返回失败。
forwardingServerReturnFailed	0x400010FA	转发服务返回失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
storageServer ReturnFailed	0x400010FB	存储服务返回失败。
cloudAnalysisServerRet urnFailed	0x400010FC	云分析服务返回失败。
modelEmpty	0x400010FD	获取到的模型为空。
mainAndBackupNodeC annotModifyManagem entNetworkInterfaceIP	0x400010FE	不允许修改主备节点的管理网口 IP 地址。
IDTooLong	0x400010FF	ID 过长。
pictureCheckFailed	0x40001100	图片删除失败。
pictureModelingFailed	0x40001101	图片建模失败。
setCloudAnalsisDefault ProvinceFailed	0x40001102	设置云分析服务默认省份失败。
InspectionAreasNumbe rExceedLimit	0x40001103	检测区域个数已达上限。
picturePixelsTooLarge	0x40001105	图片分辨率过高。
picturePixelsTooSmall	0x40001106	图片分辨率过低。
storageServiceIPEmpty	0x40001107	存储服务器 IP 地址为空。
bigDataServerRequestI nfoBuildFail	0x40001108	大数据服务请求信息组建失败。
analysisiTimedOut	0x40001109	分析超时。
high- performanceModeDisa bled.	0x4000110A	请开启高性能模式。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
configuringUPSMonitoringServerTimedOut	0x4000110B	UPS 检测服务器配置超时。请检查 IP 地址。
cloudAnalysisRequestInformationBuildFailed	0x4000110C	云分析服务请求信息组建失败。
cloudAnalysisResponseInformationParseFailed	0x4000110D	云分析服务响应信息解析失败。
allCloudAnalysisInterfaceFailed	0x4000110E	调用云分析接口失败。
cloudAnalysisModelCompareFailed	0x4000110F	云分析模型比对失败。
cloudAnalysisFacePictureQualityRatingFailed	0x40001110	云分析人脸质量评分失败。
cloudAnalysisExtractFeaturePointsFailed	0x40001111	云分析特征点提取失败。
cloudAnalysisExtractPropertyFailed	0x40001112	云分析属性提取失败。
getAddedNodeInformationFailed	0x40001113	获取数据分析服务器上已添加节点信息失败。
noMoreAnalysisUnitsAdded	0x40001114	数据分析服务器数量已达上限。
detectionAreaInvalid	0x40001115	检测区域无效。
shieldAreaInvalid	0x40001116	屏蔽区域无效。
noMoreShieldAreasAdded	0x40001117	屏蔽区域个数已达上限。
onlyAreaOfRectangleShapeAllowed	0x40001118	检测区域仅支持绘制矩形。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
numberReachedLLimit	0x40001119	数量已达上限。
wait1~3MinutesGetIPAfterSetupDHCP	0x4000111A	配置 DHCP 后请等待 1 到 3 分钟才能获取 IP 地址。
plannedTimeMustbeHalfAnHour	0x4000111B	计划时间必须为整半小时。
oneDeviceCannotBuildCluster	0x4000111C	一台设备无法组建主备集群。
updatePackageFileNotUploaded	0x4000111E	升级包文件未上传。
highPerformanceTasksNotSupportDrawingDetectionRegions	0x4000111F	高性能模式不支持绘制检测区域。
controlCenterIDDoesNotExist	0x40001120	控制中心 ID 不存在。
regionIDDoesNotExist	0x40001121	区域 ID 不存在。
licensePlateFormatError	0x40001122	车牌格式错误。
managementNodesNotSupportThisOperation	0x40001123	不支持该操作。
searchByPictureResourceNotConfiged	0x40001124	未配置以图搜图条件。
videoFileEncapsulationFormatNotSupported	0x40001125	不支持该录像文件封装格式。
videoPackageFailure	0x40001126	录像转封装失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
videoCodingFormatNotSupported	0x40001127	不支持该视频编码格式。
monitorOfDeviceArmingdeleteArmingInfo	0x40001129	监控点已布控。请先删除布控信息后再操作。
getVideoSourceTypeFailed	0x4000112A	获取视频源类型失败。
smartRulesBuildFailed	0x4000112B	组建智能规则失败。
smartRulesParseFailed	0x4000112C	解析智能规则失败。
timeRulesBuildFailed	0x4000112D	组建时间规则失败。
timeRulesParseFailed	0x4000112E	解析时间规则失败。
monitoInfoInvalid	0x4000112F	监控点信息无效。
addingFailedVersionMismatches	0x40001130	添加失败。设备版本不匹配。
theInformationReturnedAfterCloudAnalysisIsEmpty	0x40001131	云分析服务返回信息为空。
selectingIpAddressOfHostAndSpareNodeFailedCheckTheStatus	0x40001132	主备节点 IP 选取失败。请检查主备节点状态。
theSearchIdDoesNotExist	0x40001133	检索 ID 不存在。
theSynchronizationIdDoesNotExist	0x40001134	同步 ID 不存在。
theUserIdDoesNotExist	0x40001136	用户 ID 不存在。
theIndexCodeDoesNotExist	0x40001138	索引码不存在。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
theControlCenterIdDoesNotExist	0x40001139	控制中心 ID 不存在。
theAreaIdDoesNotExist	0x4000113A	区域 ID 不存在。
theArmingLinkageIdDoesNotExist	0x4000113C	布控关系 ID 不存在。
theListLibraryIdDoesNotExist	0x4000113D	名单库 ID 不存在。
invalidCityCode	0x4000113E	城市码无效。
synchronizingThePasswordOfSpareServerFailed	0x4000113F	备机系统密码同步失败。
editingStreamingTypesNotSupported	0x40001140	不支持修改取流类型。
switchingScheduledTaskToTemporaryTasksNotSupported	0x40001141	不支持计划任务切换到临时任务。
switchingTemporaryTaskToScheduledTasksNotSupported	0x40001142	不支持临时任务切换到计划任务。
theTasksNotDispatchedOrItIsUpdating	0x40001143	任务未调度或正在更新。
thisTaskDoesNotExist	0x40001144	云分析服务不存在该任务。
duplicatedSchedule	0x40001145	计划时间段不能重合。
continuousScheduleWithSameAlgorithmTypeShouldBeMerged	0x40001146	算法类型相同且连续的计划时间需合并。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
invalidStreamingTimeRange	0x40001147	取流时间范围无效。
invalidListLibraryType	0x40001148	名单库类型错误。
theNumberOfMatchedResultsShouldBeLargerThan0	0x40001149	查询结果数应大于 0。
invalidValueRangeOfSimilarity	0x4000114A	相似度区间错误。
invalidSortingType	0x4000114B	排序类型无效。
noMoreListLibraryCanBeLinkedToTheDevice	0x4000114C	名单数超出单机限制上限。
InvalidRecipientAddressFormat	0x4000114D	结果接收方地址格式错误。
creatingClusterFailedTheDongleIsNotPluggedIn	0x4000114E	组建集群前先插入加密狗。
theURLIsTooLong	0x4000114F	任务未配置计划。
noScheduleIsConfiguredForTheTask	0x40001150	计划任务未计划时间。
theDongleIsExpired	0x40001151	加密狗已过期。
dongleException	0x40001152	加密狗异常。
invalidKey	0x40001153	授权服务 key 值无效。
decryptionFailed	0x40001154	授权服务解密失败。
encryptionFailed	0x40001155	授权服务加密失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
AuthorizeServiceResponseError	0x40001156	授权服务响应异常。
incorrectParameter	0x40001157	授权服务参数错误。
operationFailed	0x40001158	授权服务器操作失败。
noAnalysisResourceOrNoDataInTheListLibrary	0x40001159	云分析无资源或名单库数据为空。
calculationException	0x4000115A	计算节点异常。
allocatingList	0x4000115B	正在分配名单。
thisOperationIsNotSupportedByTheCloudAnalytics	0x4000115C	云分析服务不支持该操作。
theCloudAnalyticsIsInterrupted	0x4000115D	云分析服务操作中断。
theServiceIsNotReady	0x4000115E	服务未就绪。
searchingForExternalApiFailed	0x4000115F	查询外部接口失败。
noOnlineNode	0x40001160	在线节点数为空。
noNodeAllocated	0x40001161	分配的节点数为空。
noMatchedList	0x40001162	查询名单数为空。
allocatingFailedTooManyFacePictureLists	0x40001163	分配失败。大数据服务名单数过多。
searchIsNotCompletedSearchAgain	0x40001164	当前查询未完成。请重新检索。
allocatingListIsNotCompleted	0x40001165	分配名单未完成。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
searchingForCloudAnalyticsResultsFailed	0x40001166	查询云分析服务超时。
noDataOfTheCurrentLibraryFound	0x40001167	当前库无数据。请先确保 Hbase 里有数据。
noFacePictureLibraryIsArmed	0x40001168	当前人脸大数据没有布控库。
noAvailableDataSlicingVersionInformationArmedFirstAndSliceTheData	0x40001169	基准版本信息无效。
duplicatedOperationDataSlicingIsExecuting	0x4000116A	打基准失败。不能重复触发打基准。
slicingDataFailedNoArmedFacePictureLibrary	0x4000116B	打基准失败。人脸大数据中没有相应的布控信息。
GenerateBenchmarkFileFailedSlicingAgain	0x4000116C	生成基准文件失败。Slice again.
NonprimaryNodesProhibitedFromSlicingData	0x4000116D	非集群主节点无法进行打基准操作。
NoReadyNodeToClusterServers	0x4000116E	组建集群失败。No ready node.
NodeManagementServicesOffline	0x4000116F	节点管理服务不在线。
theCamera(s)OfTheControlCenterAreAlreadyArmed.DisarmThemFirst	0x40001170	该控制中心下已有监控点关联布控。请先解除后再重新尝试。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
theCamera(s)OfTheAreaAreAlreadyArmed.DisarmThemFirst	0x40001171	该区域下已有监控点关联布控。请先解除后再重新尝试。
configuringHigh-frequencyPeopleDetectionFailed	0x40001172	高频检测配置失败。
searchingForHigh-frequencyPeopleDetectionLogsFailed.	0x40001173	高频人员检测事件记录查询失败。
gettingDetailsOfSearchedHigh-frequencyPeopleDetectionLogsFailed.	0x40001174	高频人员检测查询搜索结果的详细信息失败。
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	该控制中心下已有监控点关联布控。
disarmingFailedTheCamerasNotArmed	0x40001177	解除布控失败。监控点未布控。
noDataReturned	0x40001178	大数据服务返回结果为空。
preallocFailure	0x40001179	预分配算法资源失败。
overDogLimit	0x4000117A	配置失败。预分配资源数已达加密狗上限。
analysisServicesDoNotSupport	0x4000117B	云分析服务不支持该功能。
commandAndDispatchServiceError	0x4000117C	云分析调度服务错误。
engineModuleError	0x4000117D	云分析引擎模块错误。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
streamingServiceError	0x4000117E	云分析取流组件错误。
faceAnalysisModuleError	0x4000117F	云分析人脸分析模块错误。
vehicleAnalysisModuleError	0x40001180	云分析车辆图片分析模块错误。
videoStructuralAnalysisModuleError	0x40001181	云分析视频结构化模块错误。
postprocessingModuleError	0x40001182	云分析后处理模块错误。
frequentlyAppearedPersonAlarmsAlreadyConfiguredForListLibrary	0x40001183	黑名单库已布控高频报警。
creatingListLibraryFailed	0x40001184	创建名单库失败。
invalidIdentityKeyOfListLibrary	0x40001185	名单库 identityKey 无效。
noMoreDevicesCanBeArmed	0x40001186	监控点数目已达上限。
settingAlgorithmTypeForDeviceFailed	0x40001187	分配任务资源失败。
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	设置高频报警失败。
invalidSearchCondition	0x40001189	结果无效。
noMoreTasksCanBeAdded	0x4000118A	任务数量达到上限

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
theTaskIsNotCompleted	0x4000118B	任务未完成。
resourceOverRemainLimit	0x4000118C	预分配资源数已达上限。
frequentlyAppearedPersonAlarms AlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	监控点已配置高频报警。请先删除布控信息后再操作。
notEnoughStorageAvailable	0x4000121E	可用存储空间不足。
switchtimedifflesslimit	0x4000123B	开机和关机之间的时间差需要小于 10 分钟。
noData	0x4000124B	导出设备无数据。
TasksNumOverRemainLimit	0x40001280	创建的任务数超过剩余允许配置的任务数量。
noMorePeopleNumChangeRulesAdded	0x4000128A	人数变化规则个数已达上限。
noMoreViolentMotionRulesAdded	0x4000128D	剧烈运动规则个数已达上限。
noMoreLeavePositionRulesAdded	0x4000128E	离岗规则个数已达上限。
SMRDiskNotSupportRaid	0x40001291	SMR 硬盘限制，不支持 RAID (性能原因)。
OnlySupportHikAndCustomProtocol	0x400012a3	目前仅支持通过设备网络 SDK 和自定义协议添加 IPv6 相机。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
vehicleEnginesNoResource	0x400012a6	车辆引擎资源不足。
noMoreRunningRulesAdded	0x400012a9	人员奔跑规则个数已达上限。
noMoreGroupRulesAdded	0x400012aa	人员聚集规则个数已达上限。
noMoreFailDownRulesAdded	0x400012ab	人员倒地规则个数已达上限。
noMorePlayCellphoneRulesAdded	0x400012ac	玩手机规则个数已达上限。
fireEscapeDetectionDurationNotLessThanAnalysisEnginesAutoSwitchTime	0x400012B3	消防通道物品滞留时间大于等于其关联的引擎轮询时间。
analysisEnginesAutoSwitchTimeNotGreaterThanFireEscapeDetectionDuration	0x400012B4	引擎的轮询时间小于等于其关联通道的消防通道物品滞留时间。
leavePositionDelayNotLessThanAnalysisEnginesAutoSwitchTime	0x400012B5	离睡岗时间大于等于其关联的引擎轮询时间。
analysisEnginesAutoSwitchTimeNotGreaterThanLeavePositionDelay	0x400012B6	引擎的轮询时间小于等于其关联通道的离睡岗时间。
cityManagementAlarmIntervalTimeNotLessThanAnalysisEnginesAutoSwitchTime	0x400012B7	智慧城管报警间隔时间大于等于其关联的引擎轮询时间。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
analysisEnginesAutoSwitchTimeNotGreaterThanCityManagementAlarmIntervalTime	0x400012B8	引擎的轮询时间小于等于其关联通道的智慧城管报警间隔时间。
taskBusy	0x400012B9	任务正在执行其他操作，无法响应本次操作。 (比如正在暂停操作，无法查询进度或者重新开始)
filePathIsEmpty	0x400012BA	路径为空。
filePathLengthIsTooLong	0x400012BB	路径太长。
filePathIsNotExist	0x400012BC	路径不存在。
noMoreUnregisteredStreetVendorRulesAdded	0x400012BD	非法摆摊 (游摊小贩) 规则个数已达上限。
noMoreStallOutsideShopRulesAdded	0x400012BE	店外经营规则个数已达上限。
noMoreStallOccupyingRoadRulesAdded	0x400012BF	占道经营规则个数已达上限。
noMoreIllegalHeapRulesAdded	0x400012C0	乱堆物堆料规则个数已达上限。
noMoreIllegalParkingofNon-MotorVehicleRulesAdded	0x400012C1	非机动车乱停放规则个数已达上限。
noMoreIllegalOutdoorAdvertisementRulesAdded	0x400012C2	户外广告规则个数已达上限。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
noMorePackGarbageRulesAdded	0x400012C3	打包垃圾规则个数已达上限。
noMoreStallUnderUmbrellaRulesAdded	0x400012C4	违规撑伞规则个数已达上限。
noMoreDustbinOverflowRulesAdded	0x400012C5	垃圾箱满溢规则个数已达上限。
noMoreExposeGarbageRulesAdded	0x400012C6	暴露垃圾规则个数已达上限。
noMoreHangClothingAlongStreetRulesAdded	0x400012C7	沿街晾晒规则个数已达上限。
ruleEventTypeDuplicate	0x400012C8	规则事件类型重复。
noCDsInCakeBoxwithEmptyCDs	0x400012CC	入盘桶无光盘。
deviceIsPrinting	0x400012CD	设备正在打印。
CDWarehouseDoorIsOpen	0x400012CE	光盘桶已开启。
plateListIsBindingChannel	0x400012D0	车牌库已关联通道，请解除关联后重试。
licensePlateNumWillOverflow	0x400012D1	车牌个数将超过上限。
plateListIsOperating	0x400012D2	车牌数据正在处理，不允许操作。
channelNotReady	0x400012D3	智能通道未开启或者初始化完成，请稍后进行规则配置。
frequentFormatDisk	0x400012D4	格式化操作太频繁，请稍后再试。
getFileTimeError	0x400012D5	获取录像文件时间失败。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
noMoreRetentionRulesAdded	0x400015ad	人员滞留规则个数已达上限。
noMoreSleepOnDutyRulesAdded	0x400015ae	睡岗规则个数已达上限。
PlateListIDNotExist	0x400015BB	车牌库 ID 不存在。
onlyForHikProtocolAccess	0x400015B	非海康协议接入，不支持高级移动侦测。
onlyH264AndH265	0x400015BD	编码格式错误，只支持 H264 和 H265。
overResolutionLimited	0x400015BE	分辨率超限。
PlateIDNotExist	0x400015BF	车牌 ID 不存在。
noMoreVMDHumanVehicle	0x400015C0	高级移动侦测路数已达上限。
taskIDArrayOverflow	0x400015C1	任务数组超过能力范围。
polygonNotAllowedCrossing	0x400015C2	多边形不允许交叉。
coverTemplateBinded	0x400015D3	模板已经被绑定，请更换其他模板绑定。
noEncodedResource	0x400015E8	设备可用编码资源不足。
fileNameTooLong	0x400015EF	文件名长度超限。
insufficientInquestFileSpace	0x400015F0	审讯文件存储空间不足。
videoPlayerFileSizeExceedLimit	0x400015F1	上传的播放器大小超限。
builtInFileName	0x400015F2	文件名不合法，跟设备内部文件名冲突。
NotAssociatedWithOwnChannel	0x400019C1	未关联自身通道。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
AITargetBPCaptureFail	0x400019C5	AI 目标比对基准图片抓拍失败。
AITargetBPToDSPFail	0x400019C6	AI 目标比对基准图片下发 DSP 失败。
AITargetBPDuplicateName	0x400019C7	AI 目标比对基准图片名称重复。
audioFileNameWrong	0x400019D0	音频文件名称错误。
audioFileImportFail	0x400019D1	音频文件导入失败。
audioChannelInProgress	0x400015D4	音频通道正忙。
recommendAsyncSearch	0x400015F3	数据范围较大, 请采用异步查询
trajectoryDataSetInUse	0x400015F6	人员轨迹训练数据正在使用中
trajectoryDataSetExists	0x400015F5	人员轨迹训练数据已存在
NonOperationalStandbyMachine	0x400019F0	非可操作热备机。
MaximumNumberOfDevices	0x400019F1	设备数目达到上限。
StandbyMmachineCannotBeDeleted	0x400019F2	本热备机不可删除。
noClientCertificate	0x40002036	未安装用户证书。
noCACertificate	0x40002037	未安装 CA 证书。
authenticationFailed	0x40002038	认证证书失败。请检查证书。
clientCertificateExpired	0x40002039	用户证书已到期。
clientCertificateRevocation	0x4000203A	用户证书已回收。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
CACertificateExpired	0x4000203B	CA 证书已到期。
CACertificateRevocation	0x4000203C	CA 证书已回收。
connectFail	0x4000203D	连接失败。
loginNumExceedLimit	0x4000203F	用户登录数已达上限。
formattingFailed	0x40002056	格式化硬盘失败。请重试。
encryptedFormattingFailed	0x40002057	格式化已加密硬盘失败。
wrongPassword	0x40002058	校验 SD 卡密码失败。密码错误。
audiolsPlayingPleaseWait	0x40002067	正在播报，请稍候。
twoWayAudioInProgressPleaseWait	0x40002068	正在语音对讲，请稍候。
calibrationPointNumFull	0x40002069	标定点个数已达上限。
completeTheLevelCalibrationFirst	0x4000206A	水平标定未设置。
completeTheRadarCameraCalibrationFirst	0x4000206B	雷达球机标定未设置。
HBDLibNumOverLimit	0x40002092	人体库已达上限
theShieldRegionError	0x40002093	设置失败。需绘制被屏蔽物体所在的地面区域。
theDetectionAreaError	0x40002094	设置失败。需绘制检测范围所在的地面区域。
invalidLaneLine	0x40002096	保存失败，车道线不符合要求。
pointsOnStraightLine	0x4000209C	标定失败，标定点不能在一条直线上。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
TValueLessThanOrEqualZero	0x4000209D	标定失败，标定点 T 值需大于 0。
NotSupportWithVideoTask	0x400020F3	通道已开启视频结构化，不支持本功能。
noDetectionArea	0x400050df	配置失败，没有设置检测区域。
storSerDownloadFileFailed	0x40006000	从存储服务上下载文件失败
storSerDownloadFileSizeZero	0x40006001	从存储服务上下载文件大小为 0
illegalTaskID	0x40006002	任务 ID 不合法
storSerNotConfig	0x40006003	存储服务未配置
armingFailed	0x40008000	布控失败。
disarmingFailed	0x40008001	撤防失败。
clearAlarmFailed	0x40008002	清除报警信息失败。
bypassFailed	0x40008003	旁路失败。
bypassRecoverFailed	0x40008004	旁路恢复失败。
outputsOpenFailed	0x40008005	开启继电器失败。
outputsCloseFailed	0x40008006	关闭继电器失败。
registerTimeOut	0x40008007	注册超时。
registerFailed	0x40008008	注册失败。
addedByOtherHost	0x40008009	外设已被其他报警主机添加。
alreadyAdded	0x4000800A	已添加外设。
armedStatus	0x4000800B	系统处于布控状态。
bypassStatus	0x4000800C	已设为旁路状态。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
zoneNotSupport	0x4000800D	防区内不支持该操作。
zoneFault	0x4000800E	防区处于预设状态。
pwdConflict	0x4000800F	密码冲突。
audioTestEntryFailed	0x40008010	开启音频测试模式失败。
audioTestRecoveryFailed	0x40008011	关闭音频测试模式失败。
addCardMode	0x40008012	添加卡片型号。
searchMode	0x40008013	查询模式。
addRemoterMode	0x40008014	添加遥控器模式。
registerMode	0x40008015	注册模式。
exDevNotExist	0x40008016	外设不存在。
theNumberOfExDevLimited	0x40008017	无可添加的外设。
sirenConfigFailed	0x40008018	警号设置失败。
chanCannotRepeatedBinded	0x40008019	该通道已与防区关联。
inProgramMode	0x4000801B	键盘正处于编程模式。
inPaceTest	0x4000801C	步测模式。
arming	0x4000801D	已布控。
masterSlavelsEnable	0x4000802C	从雷达不支持该操作。主从关系已生效。
forceTrackNotEnabled	0x4000802D	强制跟踪已禁用。
isNotSupportZoneConfigByLocalArea	0x4000802E	当前区域不支持设置此防区类型。
alarmLineCross	0x4000802F	报警线交叉。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
zoneDrawingOutOfRange	0x40008030	绘制的防区超出探测范围。
alarmLineDrawingOutOfRange	0x40008031	绘制的报警线超出探测范围。
hasTargetInWarningArea	0x40008032	警戒区有目标。是否启用强制布防？
radarModuleConnectFail	0x40008033	雷达模块通讯失败。
importCfgFilePasswordErr	0x40008034	导入配置文件时密码错误。
overAudioFileNumLimit	0x40008038	音频文件个数超出限制。
audioFormatIsWrong	0x4000803a	音频文件格式不对。
audioFileIsLarge	0x4000803b	音频文件过大。
beyondNumberLimit	0x4000a012	超过个数上限
subscribeTypeErr	0x4000a016	该元数据类型不支持订阅。
EISError	0x4000A01C	智能开启中，电子防抖失败。
jpegPicWithAppendDataError	0x4000A01D	抓热图失败。测温配置参数异常（反射率、距离、反射温度配置过大或过小都会导致抓热图失败）。
HDMIResolutionIllegal	0x40002040	HDMI 视频分辨率不能大于主码流和子码流的分辨率。
startAppFail	0x400020B2	开启应用程序失败。
yuvconflict	0x400020B3	原始视频流冲突。
overMaxAppNum	0x400020B4	应用程序上传数已达上限。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
alreadyExist	0x40002029	应用程序已存在。
noFlash	0x400020B5	Flash 不足。
alreadyRunning	0x40002026	应用程序正在运行。
notRunning	0x40002027	应用程序已停止。
packNotFound	0x40002028	软件包不存在。
noMemory	0x4000202A	内存不足。
hdFormatFail	0x40002049	格式化硬盘失败。
invalLicense	0x4000202B	许可证无效。
plsConfirmClassroomIsCon f	0x400020C4	操作失败, 请确认该教室是否在互动中。
forkliftFilterConflictsWi thShipDetection	0x40004006	叉车过滤与船只检测互斥, 请关闭船只检测后重试。
shipDetectionConflicts WithForkliftFilter	0x40004007	船只检测与测温功能下的叉车过滤互斥, 请关闭叉车过滤后重试。
forkliftFilterConflictsWi thWasteGasDetection	0x40004008	叉车过滤与废气检测互斥, 请关闭废气检测后重试。
wasteGasDetectionConf lictsWithForkliftFilter	0x40004009	废气检测与测温功能下的叉车过滤互斥, 请关闭叉车过滤后重试。
forkliftFilterConflictsWi thVehicleDetection	0x4000400A	叉车过滤与车辆检测互斥, 请关闭车辆检测后重试。
vehicleDetectionConfli ctsWithForkliftFilter	0x4000400B	车辆检测与测温功能下的叉车过滤互斥, 请关闭叉车过滤后重试。
sunReflectionBlurFaile dWithFusionCalibratio n	0x4000400C	开启太阳反光过滤失败, 请联系专业人员进行双光融合标定。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
emptyEventName	0x400015E0	事件描述不允许为空。
sameEventName	0x400015E1	存在相同的事件描述。
emptyEventType	0x400015E2	事件类型不允许为空。
sameEventType	0x400015E3	存在相同的事件类型。
maxEventNameReached	0x400015E4	超过最大事件描述支持数量。
hotSpareNotAllowedExternalStorage	0x400015FC	热备状态下无法开启外部存储器。
sameCustomProtocolName	0x400015FD	存在相同的协议名称。
maxPTZTriggerChannelReached	0x400015FE	PTZ 联动通道达到上限。
POSCannotAddHolidayPlan	0x400015FF	假日计划不支持 POS 事件。
eventTypesTooLong	0x40001600	事件类型太长。
eventNamesTooLong	0x40001601	事件名称太长。
PerimeterEnginesNoResource	0x40001602	周界引擎资源不足。
invalidProvinceCode	0x40001607	非法省份码。
trainAndScoreNotExecutedTogether	0x4000160A	训练任务、评分任务同时只允许执行一个。

statusCode=5

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
badXmlFormat	0x50000001	XML 格式错误。

statusCode=6

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
badParameters	0x60000001	参数错误。
badHostAddress	0x60000002	主机 IP 地址错误。
badXmlContent	0x60000003	XML 内容错误。
badIPv4Address	0x60000004	IPv4 地址错误。
badIPv6Address	0x60000005	IPv6 地址错误。
conflictIPv4Address	0x60000006	IPv4 地址冲突。
conflictIPv6Address	0x60000007	IPv6 地址冲突。
badDomainName	0x60000008	域名错误。
connectServerFail	0x60000009	连接服务器失败。
conflictDomainName	0x6000000A	域名冲突。
badPort	0x6000000B	端口数冲突。
portError	0x6000000C	端口错误。
exportErrorData	0x6000000D	导入数据失败。
badNetMask	0x6000000E	子网掩码错误。
badVersion	0x6000000F	版本不匹配。
badDevType	0x60000010	设备类型不匹配。
badLanguage	0x60000011	语言不匹配。
incorrentUserNameOrPassword	0x60000012	用户名或者密码不正确。
invalidStoragePoolOfCloudServer	0x60000013	云存储的存储池无效。未配置存储池或存储池 ID 错误。
noFreeSpaceOfStoragePool	0x60000014	存储池空间不足。
riskPassword	0x60000015	密码安全性低。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
UnSupportCapture	0x60000016	Smart264 开启后, 无法抓拍分辨率为 4096*2160 或 3072*2048 的图片。
userPwdLenUnder8	0x60000023	密码至少需要包含以下两种: 数字, 大小写字母, 特殊字符。
userPwdNameSame	0x60000025	用户名和密码相同。
userPwdNameMirror	0x60000026	密码不能是用户名倒写。
beyondARGSRangeLimit	0x60000027	参数值超出限制。
DetectionLineOutOfDetectionRegion	0x60000085	规则线不在区域内。
DetectionRegionError	0x60000086	规则区域错误。请确保规则区域为凸多边形。
DetectionRegionOutOfCountingRegion	0x60000087	规则区域必须用红框标记。
PedalAreaError	0x60000088	踏板区域必须置于规则区内。
DetectionAreaABError	0x60000089	探测区 A 和 B 必须在规则框内。
ABRegionCannotIntersect	0x6000008A	区域 A 和 B 不能重合。
customHBPIDError	0x6000008b	错误的自定义人体库图片 ID。
customHBPIDRepeat	0x6000008c	重复的自定义人体图片 ID。
dataVersionsInHBDLibMismatches	0x6000008d	人体库中数据库版本不匹配。
invalidHBPID	0x6000008e	人体数据 PID 无效。
invalidHBDID	0x6000008f	人体库 ID 无效。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
humanLibraryError	0x60000090	人体库数据错误。
humanLibraryNumError	0x60000091	超过人体库最大个数。
humanImagesNumError	0x60000092	超过人体图片最大张数。
noHumanInThePicture	0x60000093	建模失败，未识别到人体。
analysisEnginesNoResourceError	0x60001000	无分析引擎资源。
analysisEnginesUsageExcced	0x60001001	引擎使用率超出负荷。
PicAnalysisNoResourceError	0x60001002	无对应智能分析引擎支持图片二次识别。
analysisEnginesLoadingError	0x60001003	分析引擎正在初始化。
analysisEnginesAbnormaError	0x60001004	分析引擎异常。
analysisEnginesFacelibImporting	0x60001005	图片正在导入人脸库。修改分析引擎参数失败。
analysisEnginesAssociatedChannel	0x60001006	分析引擎已关联通道。
smdEncodingNoResource	0x60001007	移动侦测编码资源不足。
smdDecodingNoResource	0x60001008	移动侦测解码资源不足。
diskError	0x60001009	硬盘错误。
diskFull	0x6000100A	硬盘已满。
facelibDataProcessing	0x6000100B	正在处理人脸库数据。
capturePackageFailed	0x6000100C	抓包失败。
capturePackageProcessing	0x6000100D	正在抓包。
noSupportWithPlaybackAbstract	0x6000100E	不支持该功能。已开启视频摘要回放。
insufficientNetworkBandwidth	0x6000100F	网络带宽不足。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
tapeLibNeedStopArchive	0x60001010	请先停止磁带库归档。
identityKeyError	0x60001011	交互口令错误。
identityKeyMissing	0x60001012	交互口令缺失。
noSupportWithPersonDensityDetect	0x60001013	不支持该功能。已开启人员密度检测。
ipcResolutionOverflow	0x60001014	IPC 分辨率超限。
ipcBitrateOverflow	0x60001015	IPC 码率超限。
tooGreatTimeDifference	0x60001016	设备和服务器之间时间差过大。
noSupportWithPlayback	0x60001017	不支持该功能。已开启回放。
channelNoSupportWithSMD	0x60001018	不支持该功能。已开启移动侦测。
channelNoSupportWithFD	0x60001019	不支持该功能。已开启人脸抓拍。
illegalPhoneNumber	0x6000101A	电话号码无效。
illegalCertificateNumber	0x6000101B	证件号码无效。
linkedCameraOutLimit	0x6000101C	连接监控点超时。
achieveMaxChannelLimit	0x6000101E	通道数已达上限。
humanMisInfoFilterEnabledChannelNumError	0x6000101F	开启人体去误报通道个数达到上限。
humanEnginesNoResource	0x60001020	人体分析引擎资源不足。
taskNumberOverflow	0x60001021	任务数已达上限。
collisionTimeOverflow	0x60001022	碰撞时间已达上限。
invalidTaskID	0x60001023	任务 ID 无效。
eventNotSupport	0x60001024	不支持该事件订阅。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
invalidEZVIZSecretKey	0x60001034	Hik-Connect 验证码无效。
needDoubleVerification	0x60001042	需双重认证。
noDoubleVerificationUser	0x60001043	无双重认证用户。
timeSpanNumOverLimit	0x60001044	时间段数量已达上限。
channelNumOverLimit	0x60001045	通道数已达上限。
noSearchIDResource	0x60001046	查询 ID 资源不足。
oneKeyExport	0x60001047	正在进行导出操作，请稍后再试。
oneKeyRemove	0x6000104e	正在进行删除操作，请稍后再试。
noSupportDeleteStrangerLib	0x60001051	不支持陌生人名单库检测。
noSupportCreateStrangerLib	0x60001052	不支持创建陌生人名单库。
behaviorAnalysisRuleInfoError	0x60001053	行为分析规则参数错误。
safetyHelmetParamError	0x60001054	安全帽参数错误。
OneChannelOnlyCanBindOneEngine	0x60001077	一个通道只能绑定一个引擎。现在已经绑定了一个引擎了，无法再次绑定。
imagePositionError	0x6000107B	图片位置坐标超出范围。
textPositionError	0x6000107C	文本位置坐标超出范围。
LogDiskNotSetReadOnlyInGroupMode	0x60001100	硬盘组中的日志硬盘无法设为只读。
LogDiskNotSetRedundancyInGroupMode	0x60001101	硬盘组中的日志硬盘无法设为备份。
AudioFileNameDuplicate	0x60001135	音频文件名称重复。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
CurrentAudioFileAIRuleInUseAIreadyDelete	0x60001136	当前音频文件使用的 AI 规则联动已删除。
TransitionUseEmmc	0x60002000	开启设备失败。EMMC 使用过度。
AdaptiveStreamNotEnabled	0x60002001	未启动码流自适应功能。
AdaptiveStreamAndVariableBitrateEnabled	0x60002002	码流自适应和动态码率不能同时启用。
noSafetyHelmetRegion	0x60002023	安全帽检测区域未配置。
unclosedSafetyHelmet	0x60002024	安全帽检测已启用。
width/heightRatioOfPictureError	0x6000202C	上传图片的宽高比最大为 2:1，最小为 1:2。
PTZNotInitialized	0x6000202E	云台未初始化。
PTZMoving	0x6000202F	云台运动中。
PTZLocked	0x60002030	云台已锁定。
deployExceedMax	0x60006018	已达到最大布防连接数。
detectorTypeMismatch	0x60008000	探测器类型不匹配。
nameExist	0x60008001	名称已存在。
uploadImageSizeError	0x60008016	上传图片的大小不能超过 5MB。
laneAndRegionOverlap	0x60008013	通道重叠。
unitConfigurationNotInEffect	/	单位参数无效。
ruleAndShieldingMaskConflict	/	规则区域与屏蔽区域重叠。
wholeRuleInShieldingMask	/	屏蔽区域的测温规则已配置。
holidayNameContainChineseOrSpecialChar	0x60001080	假日名称不支持中文和特殊字符。

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
genderValueError	0x60001081	性别取值不合法。
certificateTypeValueError	0x60001082	证件类型不合法。
personInfoExtendValueIsTooLong	0x60001083	人脸库自定义标签长度超限。
personInfoExtendValueContainsInvalidChar	0x60001084	人脸库自定义标签含有非法字符。
excelHeaderError	0x60001085	Excel 表头错误。

statusCode=7

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
rebootRequired	0x70000001	重启生效。

statusCode=8

子状态码 (subStateCode)	错误码 (errorCode)	描述 (errorMsg)
notAllSuccess	0x80000001	批量操作部分成功。
allSuccess	0x80000002	批量操作全部成功。
allFailed	0x80000003	批量操作全部失败。

B.4 功能模块化错误码

ISAPI 集成过程中返回的功能模块化错误码是根据产品功能模块进行分类的。错误码、错误码解释和问题排查建议详见下表。

公共功能模块（错误码范围：0x00000000、0x00100001 至 0x001fffff）

错误码字符串	错误码	错误码描述	问题排查建议
success	0x00000000	成功	
deviceNotActivated	0x00100001	设备没有激活。	激活设备。
deviceNoPermission	0x00100002	设备操作失败，原因是权限不够。	更新操作用户的权限。
deviceNotSupport	0x00100003	设备不支持该功能。	区分设备能力级调用对应支持的功能接口。
deviceResourceNotEnough	0x00100004	设备运行资源不足。	释放设备运行资源。
dataFormatError	0x00100005	报文格式不符合语法(JSON)要求。	
resetError	0x00100006	断电重启后恢复出厂需要重新激活，原因是 reset 键状态异常，通常是卡住了。	
parameterError	0x00100007	参数错误。	
	0x00100100	通道不合法。	请确认该通道是否合法。
	0x00100101	加密码流不支持 NPQ 取流。	加密码流取流请更换其他取流方式。
	0x00100102	NPQ 取流路数达到上限。	请降低 npq 取流路数再尝试取流。
	0x00100103	不支持的码流类型。	请检查请求取流码流的类型。
	0x00100104	连接数超限。	请降低取流客户端数量后重新取流。
	0x00100105	带宽不足。	请减少远程取流路数。

用户功能模块（错误码范围：0x00200001 至 0x002fffff）

错误码字符串	错误码	错误码描述	问题排查建议
passwordError	0x00200001	用户名密码错误。	确认正确的密码。
userNameNotExist	0x00200002	用户名不存在。	确认用户名是否存在，或者重新添加该用户名。
userNameLocked	0x00200003	用户名被锁定。	等待设备解锁。
userNumLimited	0x00200004	用户登陆上限。	注销登陆。
lowPrivilege	0x00200005	没有足够的权限进行此操作。	<p>用户类操作的，按以下措施排查：</p> <ul style="list-style-type: none"> • 无法删除自身 • 无法修改自身用户等级或权限 • 用户权限低，无法获取高权限用户信息 • 用户无法提升低权限用户等级及权限 <p>其他类操作的，按以下措施排查：无法进行用户权限配置外的操作，检查自身的用户类型和权限；若还无法确认的，信息获取后联系研发。</p>
incorrentUserNameOrPassword	0x00200006	用户名或者密码不正确。	检查配置的用户名和密码是否对应，若不确定的，联系管理员重新配置用户名或密码；若是管理员密码遗忘的，可对设备进行密码重置
riskPassword	0x00200007	风险密码	密码复杂度过低，重新修改密码。

错误码字符串	错误码	错误码描述	问题排查建议
passwordMustContainMorethan8Characters	0x00200008	密码长度需要大于等于 8。	检查配置的密码长度是否大于等于 8，若不是重新修改密码
passwordLenNoMoreThan16	0x00200009	密码长度不能大于 16。	检查配置的密码长度是否大于 16，若是重新修改密码
adminUserNotAllowedModify	0x0020000a	无法修改 admin 信息。	检查修改的用户是否是 admin。admin 用户名、权限等信息无法被修改。
confirmPasswordError	0x0020000b	修改用户信息时，确认密码错误。	检查确认密码。
passwordMustContainMorethan2Types	0x0020000c	密码需要包含数字/小写/大写/特殊字符任意两种以上。	检查密码配置是否满足要求 (密码需要包含数字/小写/大写/特殊字符任意两种以上)。
passwordContainUserName	0x0020000d	非法的密码，密码不能包含用户名。	检查密码是否包含用户名。
userPwdNameMirror	0x0020000e	密码不能是用户名倒写。	检查密码包含用户名倒写。

时间功能模块 (错误码范围 : 0x00300001 至 0x003fffff)

错误码字符串	错误码	错误码描述	问题排查建议
manualAdjustmentFailed	0x00300001	设备校时失败。	
NTPError	0x00300002	NTP 服务地址不生效。	确认有效的 NTP 服务地址。
timeFormatError	0x00300003	校时过程中下发时间格式错误。如 ISO 8601 时间格式的本地时间为 : 2018-02-	报文格式错误或时间内容格式错误。

错误码字符串	错误码	错误码描述	问题排查建议
		01T19:54:04, 实际下发为 : 2018-02-01 19:54:04。	
beyondTimeRange Limit	0x00300004	校时时间不在设备端支持的时间范围内。如设备端支持校时范围为 : 2000-01-01T00:00:00~2037-12-31T23:59:59, 实际下发 2038-01-01T00:00:00。	查看设置的时间与设备能力进行比较, 看是否超范围。
endtimeEarlierThan BeginTime	0x00300005	有效期开始时间不能晚于结束时间 (如人员有效期, 起始时间必须小于结束时间等)。	查看比较起始时间与结束时间是否合法。

网络功能模块（错误码范围：0x00400001 至 0x004fffff）

错误码字符串	错误码	错误码描述	问题排查建议
domainNameParse Failed	0x00400001	域名解析失败。	
PPPOEConnectedFailed	0x00400002	PPOE 联网失败。	
FTPConnectedFailed	0x00400003	FTP 服务器掉线。	
deviceIPConflicted	0x00400004	IP 地址冲突。	
libraryConnectedFailed	0x00400005	图像视频库掉线未连接。	
fileUploadFailed	0x00400006	文件 (包括图片) 上传失败。	排查网络连接, 如没有问题, 则联系售后。

错误码字符串	错误码	错误码描述	问题排查建议
storSerDownloadFileFailed	0x00400007	1. 从存储服务上下载文件(包括图片)失败。 2. 文件(包括图片)下载失败。	排查网络连接, 如没有问题, 则联系售后。
storSerDownloadFileSizeZero	0x00400008	从存储服务上下载文件(包括图片)大小为 0。	排查网络连接, 如没有问题, 则联系售后。
storSerNotConfig	0x00400009	存储服务未配置。	检查配置是否正确。
badHostAddress	0x0040000a	主机地址错误。	检查配置是否正确。
badIPv4Address	0x0040000b	IPv4 地址错误。	检查配置是否正确。
badIPv6Address	0x0040000c	IPv6 地址错误。	检查配置是否正确。
conflictIPv4Addresses	0x0040000d	IPv4 地址冲突。	检查网络中 IPv4 地址配置情况。
conflictIPv6Addresses	0x0040000e	IPv6 地址冲突。	检查网络中 IPv6 地址配置情况。
badDomainName	0x0040000f	错误的域名	检查配置是否正确。
connectServerFail	0x00400010	连接服务器失败。	检查网络是否通畅, 检查配置是否正确。
conflictDomainName	0x00400011	域名冲突	检查配置是否正确。
badPort	0x00400012	端口冲突	检查配置是否正确。
portError	0x00400013	端口错误	检查配置是否正确。
badNetMask	0x00400014	子网掩码错误	检查配置是否正确。
badVersion	0x00400015	版本不匹配	检查版本是否正确。
badDns	0x00400016	错误的 DNS	检查配置是否正确。
badMTU	0x00400017	错误的 MTU	检查配置是否正确。

错误码字符串	错误码	错误码描述	问题排查建议
badGateway	0x00400018	错误的网关	检查配置是否正确。
urlDownloadFail	0x00400019	URL 下载失败。	检查网络及 URL 是否正确。
deployExceedMax	0x0040001a	布防超过最大连接数。	确认最大支持布防数已及已经布防数。

维护功能模块 (错误码范围 : 0x00500001 至 0x005fffff)

错误码字符串	错误码	错误码描述	问题排查建议
upgradeXMLFormatError	0x00500001	升级的请求 xml 格式错误	确认升级文件是否有误, 若升级文件正确, 采用本地升级
upgradeContentError	0x00500002	升级的请求内容错误	确认升级文件是否有误, 若升级文件正确, 采用本地升级
noUpgradePermission	0x00500003	对应用户无权限升级	更换 admin 用户进行尝试或请求 admin 配置高级操作权限
upgrading	0x00500004	升级进行中	已经在升级中, 等待上一次升级完成
receiveUpgradePackageError	0x00500005	接收升级包出错	请排查网络是否异常
upgradePackageLanguageMismatch	0x00500006	升级包语言不匹配	请确认升级包语言类型、设备的语言类型
upgradePackageMismatch	0x00500007	升级包与设备类型不匹配	请确认升级包设备类型、设备类型
OEMCodeMismatch	0x00500008	定制客户的 oemcode 不匹配, 升级包错误。	请联系售后, 获取正确的升级包

错误码字符串	错误码	错误码描述	问题排查建议
versionMismatch	0x00500009	升级的版本不匹配	请联系售后，获取正确支持的升级包版本
upgradeHalfFailed	0x0050000c	设备升级进度到一半才提示升级失败(原因是 flash 或缓存问题)。	
deviceParameterImportFailed	0x0050000d	设备参数导入失败(设备型号不一致，版本不一致，平台不一致等)。	
deviceEncryptionError	0x0050000e	升级对应型号版本的程序提示升级包不匹配(原因是设备加密有问题，通常是售后加密错)。	
SDCardFormatError	0x00500025	SD 卡格式化失败。	
SDCardLoadFailed	0x00500026	SD 卡插入，但是页面不显示。	
NASFailed	0x00500027	NAS 挂载失败。	
hardDiskError	0x00500028	硬盘异常(硬盘不存在、不兼容、加密盘、容量不支持、格式化异常、阵列异常不兼容等)	
upgradeError	0x00500030	升级错误	从 0x00500030-0x0050005f, 分配给升级使用。
upgradePackageSizeMismatch	0x00500032	已下载的升级包大小和升级指令里的大小不匹配。	升级指令里的大小是查询升级包信息时获取到的。
upgradePackageSizeExceeded	0x00500033	文件大小超过分区大小。	

错误码字符串	错误码	错误码描述	问题排查建议
domainNameParseFailedForDownload	0x00500034	下载地址域名解析失败。	
netWorkUnstable	0x00500035	网络不稳定，下载超时（超过最大重试次数）。	
digestValueMismatch	0x00500036	摘要值不匹配	
signatureVerifyFailed	0x00500037	签名校验失败。	
innerFormatError	0x00500038	升级包损坏(内部格式不对)	
memoryNotEnough	0x00500039	内存不足	
burnFailed	0x0050003a	烧录固件出错	
unknownError	0x0050003b	底层接口出现未知错误	
userCancel	0x0050003c	用户主动停止。	
systemResume	0x0050003d	升级失败，通过备份系统或最小系统恢复。	
devicePlatformMismatch	0x00500060	升级包与设备平台类型不匹配（例如，GD 平台的升级包升级到 ST 平台的设备上）。	请确认升级包平台类型。工具获取设备平台类型，再比对升级包升级类型，不同类型不允许升级
upgradeObjectNoSupport	0x00500061	升级对象不支持。	检查升级的对象，联系研发确认升级对象是否支持升级
upgradeObjectNoOnline	0x00500062	升级对象不在线。	检查升级的对象是否在线（网络升级检查网络，网络转串口检查网络和串口），通过工具获取在线状态，不在线

错误码字符串	错误码	错误码描述	问题排查建议
			的话获取不到升级包，升级不能成功
verificationFailed	0x00500063	升级包签名校验失败。	请联系售后，获取正确支持的升级包。若使用正确升级包还是报该错误，抓包并联系研发
upgradeDataErrLength	0x00500064	升级包数据长度异常。	排查网络是否异常，排查升级包是否正常
upgradeParameterWrong	0x00500065	升级包参数错误	包含 CRC 校验失败、升级包头 没找到。请联系售后，获取正确且支持的升级包版本
passwordDecodeError	0x00500066	密码解密失败。	密码解密失败，联系研发，提供相应打印和抓包
uninitialized	0x00500067	未初始化	检查配置文件是否正确，配置给设备的资源文件是否正确，检查之后还存在问题，联系研发，并提供相应资源文件
taskNotExist	0x00500068	任务不存在。	模板批量下发时，任务不存在，确认是否操作正确。需进行抓包获取对应日志，检查抓包数据是否满足协议要求，并联系研发
noMoreTasksCanBeAdded	0x00500069	任务数量达到上限。	模板批量下发未完成，又再次进行下发操作。需进行抓包获取对应日志，检查抓包的数据流程是否满足 sdk 协议，并联系研发

错误码字符串	错误码	错误码描述	问题排查建议
configOperating	0x0050006a	设备正在导入或者导出。	设备正在导入或者导出，不允许再次操作
fileFormatError	0x0050006b	文件格式错误（例如支持 excel 的，下发了 word）	文件格式错误，检查操作的文件的格式是否合规
fileContentError	0x0050006c	文件内容错误（表格项或内错误）	文件内容错误，类似表格项或内错误，检查文件内容，是否有错误的地方
importEmptyFile	0x0050006d	文件内容为空	文件内容为空，检查操作的文件，是否是一个空文件
fileSizeExceedError	0x0050006e	文件太大 包括升级包等太大（例如，人脸下发等操作时，设备对图片大小有限制，人脸图片太大时，可能会返回失败；手动抓拍时，图片过大）。	可能的原因：人脸图片超过 200KB；可排除原因后，若还未定位，再联系研发
roomNoExisted	0x0050006f	房间号冲突（已存在）	房间号冲突（已存在），获取已存在的房间号，并检查配置的房间号是否存在于已存在的房间号中。
serialNumberConflict	0x00500070	序列号冲突	检查配置的参数，是否存在一样的序列号。
deviceIdConflict	0x00500071	设备编号冲突	检查配置的设备编号，是否存在一样的编号。
roomNoStorageFull	0x00500072	号码存储空间已满	确认设备能力，满了不允许存储。清空或者删除之后再操作。

错误码字符串	错误码	错误码描述	问题排查建议
roomNoNotExist	0x00500073	房间号不存在	获取检查房间号是否存在，不存在先进行添加。
iotNumReachesLimit	0x00500074	IOT 通道数目达到上限（例如，为添加室内机或从门口机等外接设备时已达设备支持的上限）	根据当前操作对象，检查设备已有数量是否已达设备支持最大上限。
rebootRequired	0x00500075	操作生效前需要重启	操作生效前，需要手动重启，不然该操作无效。
updateDatabaseError	0x00500076	数据库更新失败	数据库更新失败，联系研发，提供相应日志，设备端排查。
searchDatabaseError	0x00500077	数据库查找失败	数据库更新失败，联系研发，提供相应日志，设备端排查。
writeDatabaseError	0x00500078	数据库写入失败	数据库更新失败，联系研发，提供相应日志，设备端排查。
deleteDatabaseError	0x00500079	数据库元素删除失败	数据库元素删除失败，联系研发，提供相应日志，设备端排查。
searchDatabaseElementError	0x0050007a	数据库元素个数查找失败	数据库元素个数查找失败，联系研发，提供相应日志，设备端排查。
	0x00500080	没有找到升级文件	请确认升级包路径是否过长或升级包路径下是否有正确的升级包。
	0x00500081	升级包引擎类型不匹配	请选择与设备引擎类型匹配的升级包。

错误码字符串	错误码	错误码描述	问题排查建议
	0x00500082	相机域名解析失败	请确认设备是否正确配置了 DNS 服务或相机域名是否有效。
	0x00500083	相机网络不可达	请确认本地网络是否可访问添加通道所在网络。

预览功能模块（错误码范围：0x00600001 至 0x006fffff）

错误码字符串	错误码	错误码描述	问题排查建议
liveViewFailed	0x00600001	预览失败，取流链接路数达到最大上限。	
	0x00600002	请求封装格式异常	请检查请求预览的封装格式。
	0x00600003	EHome2.x 开启后不支持 NPQ	ehome2.x 开启，请更换为其他预览方式。
	0x00600005	零通道不支持 NPQ 预览	零通道预览请更换其他预览方式。
	0x00600007	仅虚拟码流支持 NPQ 预览	请更换码流类型到虚拟码流。
	0x0060000A	IP 通道不在线	请确认 ip 通道在线后再尝试预览。
	0x0060000B	设备不支持转码预览	请使用其他码流类型预览。
	0x0060000C	零通道未开启	预览零通道请先将零通道开启。
	0x0060000D	转码能力超限	请降低相机的分辨率或减少转码通道数量。
	0x00600010	该通道无子码流	本通道未接入子码流，请使用主码流方式预览。

错误码字符串	错误码	错误码描述	问题排查建议
	0x00600011	设备不支持 NPQ 预览	请更换其他预览方式。
	0x00600012	NPQ 功能未开启	请先开启 NPQ 功能或更换其他预览方式。

回放功能模块（错误码范围：0x00700001 至 0x007fffff）

错误码字符串	错误码	错误码描述	问题排查建议
	0x00700001	回放失败，原因是目前前端大部分只支持 1 路回放	
	0x00700002	回放上墙的倍数不支持	减小回放倍数。
	0x00700003	回放码流推送速率过大	减小推送码流速率。
	0x00700004	回放码流编码格式设备不支持	提供设备支持编码格式码流。
	0x00700005	回放码流封装类型设备不支持	提供设备支持封装类型码流。
	0x00700007	回放解码异常（上墙问题;图像/显示/解码异常;图像卡顿/黑屏）（不支持的码流类型、预览卡顿、解码异常、音频解码异常、花屏）	
	0x00700008	回放录像不存在或搜索失败	请尝试重新搜索或确认硬盘是否正常。
	0x00700009	回放时间参数错误	请确认搜索录像时间段正确后再尝试回放。
	0x0070000A	无效的录像类型	请选择正确的录像类型进行搜索。

错误码字符串	错误码	错误码描述	问题排查建议
	0x0070000B	无效的时间类型	请选择正确的时间类型进行搜索。
	0x0070000C	无效的事件参数	请选择正确的事件参数进行搜索。
	0x0070000D	无效的事件类型	请选择正确的事件类型进行搜索。
	0x0070000E	设备不支持智能检索	请选择非智能检索方式进行搜索。
	0x0070000F	无效的智能事件类型	请选择正确的智能事件类型进行搜索。
	0x00700010	无效的动态分析灵敏度	请选择正确的灵敏度进行录像搜索。
	0x00700011	不支持倒放	请选择正确的回放模式。
	0x00700012	无效的文件状态	请选择正确的文件状态进行搜索。
	0x00700013	无效的搜索起始位置	请使用正确的搜索起始位置进行搜索。
	0x00700014	无效的最大检索数量	请使用正确的最大检索数量检索。

抓拍功能模块（错误码范围：0x00800001 至 0x008fffff）

错误码字符串	错误码	错误码描述	问题排查建议
	0x00800001	手动抓拍失败。	

语音对讲模块（错误码范围：0x00900001 至 0x009fffff）

错误码字符串	错误码	错误码描述	问题排查建议
startFailed	0x00900001	开启对讲失败。	
codingFormatNot Match	0x00900002	对讲双方的编码格式不一致，协商失败。	平台端查看或抓包协议交互报文，分析双方协商的音频编码格式是否一致。
dialedIsBusy	0x00900003	被对讲方已处于对讲中，无法再响应对讲。	查看对方是否确实在对讲中，若不是再抓包协议交互报文，分析对方应答报文。
destinationLongNumberError	0x00900004	请求的目的长号错误。	平台端查看或抓包协议交互报文，分析长号。

录像存储模块（错误码范围：0x00a00001 至 0x00afffff）

错误码字符串	错误码	错误码描述	问题排查建议
videoSearchFailed	0x00a00001	录像查询失败。	设备没有存储资源
notFindStorageMedium	0x00a00002	没有找到存储介质。	
videoDownloadFailed	0x00a00003	录像下载失败。	
checkMemoryCard	0x40002126	查找失败，请检查存储状态是否正常。	

图像存储模块（错误码范围：0x00b00001 至 0x00bfffff）

错误码字符串	错误码	错误码描述	问题排查建议
	0x00b00001	图像查询失败	图像没有存储资源

IO 功能模块（错误码范围：0x00c00001 至 0x00cfffff）

错误码字符串	错误码	错误码描述	问题排查建议
	0x00c00001	报警输入编号无效	
	0x00c00002	报警输出编号无效	

事件功能模块（错误码范围：0x00d00001 至 0x00dfffff）

错误码字符串	错误码	错误码描述	问题排查建议
	0x00d00001	事件规则配置错误	参考操作规范配置

停车服务模块（错误码范围：0x00e00001 至 0x00efffff）

错误码字符串	错误码	错误码描述	问题排查建议
	0x00e00001	包期车已存在	包期车是以车牌创建的，所以需要查看是不是用该车牌已经创建过包期车。
	0x00e00002	车牌号参数不应该为空字符串	

通用功能模块（错误码范围：0x00f00001 至 0x00ffffff）

错误码字符串	错误码	错误码描述	问题排查建议
noMemory	0x00f00001	设备内存不足（堆空间分配失败）	linux, root 目录下 free 查看内存, free 栏目内存剩余值, 日志发给研发定位。
deviceBusy	0x00f00002	设备忙或设备无响应	日志发回研发分析； 指纹采集/人脸采集/文件下发/文件上传相关业务, 确定上一次操作是否已经执行完成

错误码字符串	错误码	错误码描述	问题排查建议
notSupport	0x00f00003	设备不支持：找不到 URL (URL 不支持)	1、wireshark 抓包，查看下发的 URL 报文. 对照 PMP 协议平台，URL 是否能搜索到。能搜索到 URL 发给研发分析定位
methodNotAllowed	0x00f00004	HTTP 方法不允许：方法和协议不匹配 如 PUT 方法协议, 下发 POST	wireshark 抓包,, 查看下发的 URL 报文，对照 PMP 协议平台，查看 URL 对应的方法。
invalidOperation	0x00f00005	接口输入指令非法操作(例如：设备不支持的通道号，不支持的硬件等下发会返回无效操作)	设备实现时，要求 errorMsg 中返回具体的错误原因
IDNotExist	0x00f00006	ID 不存在：URL 中，按协议应带 ID，但实际未带 ID	wireshark 抓包，查看 URL 携带的 ID 是否正确。
invalidID	0x00f00007	无效 ID: URL 带 ID 时，ID 超过能力集或数据格式不规范	wireshark 抓包，查看 URL 携带的 ID 是否正确； 对应 URL 有能力集的，获取 URL 对应的能力集，比对 ID 范围。
invalidIURL	0x00f00008	URL 中“？问号”后内容错误	wireshark 抓包，查看 URL 是否正确
deviceAckTimeOut	0x00f00009	设备响应超时	1. 若是与外接模块通信超时，则先查看是否外接模块掉线导致； 2. 排除问题 1 后，其他的获取日志后，联系设备研发定位。

错误码字符串	错误码	错误码描述	问题排查建议
badXmlFormat	0x00f0000a	XML 格式错误（例如：不完整，不能作为一个完成的xml 解析，如<a>,应该是<a>）	设备实现时，要求 errorMsg 中返回具体节点名称，排查时根据 errorMsg 及对应节点能力来进行
badJsonFormat	0x00f0000b	JSON 格式错误：类似 Json 格式，双引号缺失： "employeeN : "XXX"	设备实现时，要求 errorMsg 中返回具体节点名称，排查时根据 errorMsg 及对应节点能力来进行
badURLFormat	0x00f0000c	URL 格式错误：URL 格式不正确。	wireshark 抓包，查看 URL 是否正确
badXmlContent	0x00f0000d	XML 报文错误： <ul style="list-style-type: none"> • 报文只有 URL，无报文体 • 必填节点存在，无参数 • 超出参数范围限制：节点参数内容错误 	设备实现时，要求 errorMsg 中返回具体节点名称，排查时，抓包获取报文，根据 errorMsg 及对应节点能力来进行
badJsonContent	0x00f0000e	JSON 报文错误： <ul style="list-style-type: none"> • 报文只有 URL，无报文体 • 必填节点存在，无参数 • 超出参数范围限制：节点参数内容错误 	设备实现时，要求 errorMsg 中返回具体节点名称，排查时，抓包获取报文，根据 errorMsg 及对应节点能力来进行
messageParameter sLack	0x00f0000f	报文参数缺少，必选节点不存在。	设备实现时，要求 errorMsg 中返回具体节点名称，排查时根据 errorMsg 及对应节点能力来进行

错误码字符串	错误码	错误码描述	问题排查建议
invalidSearchConditions	0x00f00010	检索条件失效, 请重新检索。	对于 search 协议的, 查看 searchID 是否正确。
operObjectNotExist	0x00f00011	操作对象不存在 (对门、报警输出、报警输入相关操作, 当对象未添加时返回)	对于设备需接入门锁、读卡器的, 排查门锁是否已经接入。

控门模块（错误码范围：0x01000001 至 0x010fffff）

错误码字符串	错误码	错误码描述	问题排查建议
multiAuthenticationFailed	0x01000001	多重认证状态操作失败。	门处于多重认证状态, 且多重认证中未执行远程开门步骤时, 远程开门会返回本错误。
securityModuleOffline	0x01000002	安全门控模块掉线开门失败。	检查安全门控是否掉线。

计划模板模块（错误码范围：0x01100001 至 0x011fffff）

错误码字符串	错误码	错误码描述	问题排查建议
planNumberConflict	0x01100001	计划编号冲突 (例如周、假日、假日组、计划模版编号非法)	类似 1 个假日组中有重复假日编号, 1 个计划模板中有重复假日组编号等, 可查看平台报文来定位。
timeOverlap	0x01100002	时间段冲突 (例如, 周计划、假日计划中时间段重叠冲突)	查看报文, 看一天中的不同时段是否有时间重叠。

人员信息模块（错误码范围：0x01200001 至 0x012fffff）

错误码字符串	错误码	错误码描述	问题排查建议
deviceUserFull	0x01200001	用户存储满（权限下发时，人员数量达到设备最大存储数量）	查询设备已有的人员数量，看是否已达到最大容量
employeeNoNotExist	0x01200002	用户不存在（人员管理时，按人员修改、查询参数等业务时，若人员不存在，则可返回错误码）	按工号获取人员，看该工号是否存在
employeeNoAlreadyExist	0x01200003	用户已存在（人员管理时，按人员增加等业务时，若人员已存在，则可返回错误码）	按工号获取人员，看该工号是否存在

凭证模块（错误码范围：0x01300001 至 0x013fffff）

错误码字符串	错误码	错误码描述	问题排查建议
deviceCardFull	0x01300001	卡已满：卡片下发时，卡数量达到设备最大存储容量	查询设备已有的卡数量，看是否已达到最大容量
cardNoAlreadyExist	0x01300002	卡已存在：卡片管理时，按卡号增加等业务时，若卡片已存在，则可返回错误码	按卡号获取卡参数，看该卡号是否存在
cardNoNotExist	0x01300003	卡不存在：卡片管理时，按卡号修改、查询参数等业务时，若卡片不存在，则可返回错误码	按卡号获取卡参数，看该卡号是否存在
cardFullPerUser	0x01300004	单人卡已满：当设备对每人关联的卡数量有限制时，若对人增加新卡时，可能会返回	获取当前人员下发的卡片数量，是否已经达到设备允许的最大能力集；

错误码字符串	错误码	错误码描述	问题排查建议
moduleHandleFailed	0x01300005	模组指令处理失败	研发排查
moduleConnectFailed	0x01300006	模组连接失败	指纹模组通信线缆是否接触不良;
fingerPrintFeatureMergeFailed	0x01300007	模组或算法模板合并失败 (采集录入时调用可能发生)。	采集的指纹次数是否达到要求; 指纹采集时手指是否居中; 指纹是否有损伤; 采集的手指是否为同一枚。
moduleIllegalParameter	0x01300008	模组使用了不正确的参数	研发排查, 调用的指令参数是否合法。
fingerPrintFeatureNumInvalid	0x01300009	模组或算法指纹合并个数无效 (采集录入时调用可能发生)。	采集的指纹次数是否达到要求, 一般需要 3 次。
fingerPrintNotExist	0x0130000a	指纹不存在 (从读卡器或主机按卡号或工号获取指纹时可能发生)。	排查相关工号/卡号是否已经下载过指纹, 是否删除过指纹
notSupportDelFPBID	0x0130000b	读卡器或主机不支持按手指 ID 删除指纹。	读卡器和主机的版本, 读卡器通信协议版本 (通过版本可排查通信协议版本)。
fingerPrintLowQuality	0x0130000c	指纹质量差 (采集录入时可能发生)。	指纹采集时手指是否居中; 指纹是否有损伤。
fingerPrintFull	0x0130000d	指纹已满, 非模组端的指纹容量已满	获取读卡器参数, 查看当前存储的指纹数量;
fingerPrintAlreadyExist	0x0130000e	已存在该指纹, 指纹重复 (指纹下发时可能发生)。	查看重复的指纹关联信息, 查看重复的指纹是否为同一人;

错误码字符串	错误码	错误码描述	问题排查建议
			非同一人时联系算法研发排查模板匹配情况。
fingerPrintIDAlreadyExist	0x0130000f	主机已存在该指纹 ID，带 ID（手指编号 1-10）添加指纹时	下发的手指 ID 是否为同一个；删除指纹时是否成功。
errorFingerPrintType	0x01300010	不支持该类型指纹模板	查看录入仪和模组是否匹配。
fingerPrintFullPerUser	0x01300011	每人指纹数量达到上限	查看当前人员是否指纹已经下载满；删除指纹时是否成功。
fingerPrintNotLinkEmployeeNoOrCardNo	0x01300012	指纹未关联工号或卡号（工号或卡号字段为空）	抓包查看该指纹是否未关联工号或卡号，若是则已定位，否则提供抓包数据联系研发。
pupilDistanceTooSmall	0x01300013	图片中人脸眼间距不足：人脸下发时可能发生	DSP 返回的眼间距小于应用配置的眼间距，可将人脸图片获取到后联系研发定位。
faceDetectFailed	0x01300014	人脸检测失败：人脸下发时可能发生	算法引擎返回失败，可将人脸图片获取到后联系研发定位。
pictureFaceDetectZero	0x01300015	图片中不存在人脸图片：人脸下发时可能发生	算法引擎返回成功，但检测到人脸为 0；可先查看图片中是否存在人脸，若存在可将人脸图片获取到后联系研发定位，若不存在可直接答复。
pictureDecodingFailed	0x01300016	图片解码失败：人脸下发时可能发生	算法引擎返回失败，可将人脸图片获取到后联系研发。

错误码字符串	错误码	错误码描述	问题排查建议
picFormatError	0x01300017	图片格式设备不支持：人脸下发时可能发生	DSP 底层返回“图片格式错误”，目前 DSP 只可支持 bmp, jpg, png 格式的图片，排除此原因后，可将人脸图片获取到后联系研发。
pictureModelingFailed	0x01300018	图片建模失败：人脸下发时可能发生	算法引擎返回失败，问题定位可将人脸图片获取到后联系研发。
faceLowQuality	0x01300019	人脸质量较差：人脸采集时可能发生	一般为姿态不对、距离过近或过远等导致，可调整后重新采集；若持续失败，可拍摄采集视频后联系研发。
faceTypeError	0x0130001a	下发的人脸错误：人脸下发时可能发生	可能的原因：设备处于普通模式，下发了红外人脸；可排除原因后，若还未定位，再联系研发
deviceFaceFull	0x0130001b	人脸数已满：人脸下发时可能发生	获取设备当前人脸总数量，看是否已达最大容量，若是则已定位问题，否则联系研发。
deviceUserAlreadyExistFace	0x0130001c	人员已存在一张人脸：人脸下发时，新增人脸时可能返回	根据工号获取人脸，看该人员是否存在人脸，若是则已定位问题，否则联系研发。

错误码字符串	错误码	错误码描述	问题排查建议
faceResolutionOrRatioError	0x0130001d	人脸分辨率或宽高比异常：人脸下发时可能发生	算法引擎返回失败，可将人脸图片获取到后联系研发定位。。
faceDataNotExist	0x0130001e	用户人脸数据不存在:一般按人员查找人脸时，可能发生	基本可确定为现场未下发该人员人脸数据，可由平台检查是成工下发过或者下发后是否中间删除过，可配合设备附件信息事件（若设备支持）定位。

安全功能模块（错误码范围：0x01400001 至 0x014fffff）

错误码字符串	错误码	错误码描述	问题排查建议
decryptFailed	0x01400001	敏感信息字段解密时，解密失败；或者导入数据文件时，解密失败。	导入导出的密钥需要保持一致。
invalidCertificate	0x01400002	导入证书时导入的不是一个合法的证书文件；导入私钥时不是一个合法的私钥文件。	检查证书文件的来源及合法性，使用前可用公开的三方网站验证证书的合法性。
certificateNotmatch	0x01400003	证书不匹配，SSL/TLS 的公钥证书需要成对匹配。	上传到设备的证书，公私钥两个证书需要同时生成。
notActivated	0x01400004	设备未激活。	使用设备功能前，用 SADP 等工具激活设备。
hasActivated	0x01400005	设备已激活。	设备已激活，无需再次激活
forbiddenIP	0x01400006	IP 地址被禁止。	用户非法登录超次后，IP 锁定的禁止

错误码字符串	错误码	错误码描述	问题排查建议
bondMacAddressNotMatch	0x01400007	MAC 地址与用户不匹配。	指定用户使用特定 mac 地址才能登陆。报类似错误, 可查看该用户的参数是否绑定了特定 MAC
bondIpAddressNotMatch	0x01400008	IP 地址与用户不匹配。	指定用户使用特定 IP 地址才能登陆。报类似错误, 可查看该用户的参数是否绑定了特定 IP
badAuthorization	0x01400009	非法登录触发。	设备密码错误, 触发非法登录

广告功能模块 (错误码范围 : 0x01500001 至 0x015fffff)

错误码字符串	错误码	错误码描述	问题排查建议
materialDownloadFailed	0x01500001	素材下载失败。	网络连接是否正常。 设备是否正常运行。 看打印是什么错误。
materialNumberIsOver	0x01500002	节目表素材格式超限(例如, 一个节目表中最多包含 5 个素材, 下发的素材超过时会触发)。	检查下发的节目表中素材个数是否超过规定限制个数。

B.5 设备日志类型

报警日志

值	描述
0x1	报警输入
0x2	报警输出
0x3	移动侦测报警开始
0x4	移动侦测报警结束
0x5	视频遮挡报警开始
0x6	视频遮挡报警结束
0x7	智能报警开始
0x8	智能报警结束
0x09	交通事件报警开始
0x0a	交通事件报警结束
0x0b	网络报警开始
0x0c	网络报警结束
0x0d	网络报警恢复
0x0e	无线报警开始
0x0f	无线报警结束
0x10	人体感应报警开始
0x11	人体感应报警结束
0x12	紧急报警开始
0x13	紧急报警结束
0x16	人脸侦测报警开始
0x17	人脸侦测报警结束
0x1a	场景变更侦测报警
0x1b	离开区域侦测开始
0x1c	离开区域侦测结束
0x1d	徘徊侦测开始
0x1e	徘徊侦测结束

值	描述
0x20	越界侦测开始
0x21	越界侦测结束
0x22	区域入侵侦测开始
0x23	区域入侵侦测报警结束
0x24	音频丢失侦测
0x25	音频异常侦测
0x26	虚焦侦测开始
0x27	虚焦侦测结束
0x29	人脸侦测开始
0x2a	进入区域侦测开始
0x2b	进入区域侦测结束
0x2c	人员聚集侦测开始
0x2d	人员聚集侦测结束
0x2e	快速移动侦测开始
0x2f	快速移动侦测结束
0x30	人脸侦测结束
0x31	场景变更侦测开始
0x32	场景变更侦测结束
0x33	音频丢失侦测开始
0x34	音频丢失侦测结束
0x35	声强突变侦测开始
0x36	声强突变侦测结束
0x39	声强陡降侦测
0x3c	停车侦测开始
0x3d	停车侦测结束
0x3e	物品遗留侦测开始
0x3f	物品遗留侦测结束
0x40	物品拿取侦测开始

值	描述
0x41	物品拿取侦测结束
0x46	自动车牌识别开始
0x47	自动车牌识别结束
0x48	火点检测开始
0x49	火点检测结束
0x50	防拆侦测开始
0x51	防拆侦测结束
0x5a	船只检测
0x5b	温度预警开始
0x5c	温度预警结束
0x5d	温度报警开始
0x5e	温度报警结束
0x5f	温差报警开始
0x60	温差报警结束
0x400	防区短路报警
0x401	防区断路报警
0x402	防区异常报警
0x403	防区报警恢复
0x404	设备防拆报警
0x405	设备防拆报警恢复
0x406	读卡器防拆报警
0x407	读卡器防拆报警恢复
0x408	报警输入报警
0x409	报警输入报警恢复
0x40a	挟持报警
0x40b	内存满报警
0x40c	读卡失败报警

值	描述
0x411	POS 启用
0x412	POS 禁用

异常日志

值	描述
0x20	磁盘阵列异常
0x21	视频丢失
0x22	非法登录
0x23	硬盘满
0x24	硬盘错误
0x25	调制解调器掉线（保留）
0x26	IP 地址冲突
0x27	网络连接断开
0x28	录像异常
0x29	网络摄像机连接异常
0x2a	视频输入异常（仅适用于模拟通道）
0x2b	网络摄像机的 IP 地址冲突
0x2c	场景异常
0x2d	抓拍错误获取图片失败
0x2e	视频制式不匹配
0x2f	解码分辨率与相机分辨率不匹配
0x3a	RS-485 串口连接异常
0x3b	RS-485 串口连接恢复
0x3c	子板启动异常
0x3d	子板已插入
0x3e	子板已拔出
0x3f	温度异常

值	描述
0x41	缓冲区溢出
0x42	数字信号处理异常
0x43	断网续传录像失败
0x44	热备工作异常
0x45	启动网络摄像机 MAS 失败
0x46	IPCM 重启异常
0x47	PoE 供电异常
0x48	云存储数据上传失败
0x49	拨号异常
0x50	设备掉线
0x51	远程升级失败
0x52	音频丢失
0x53	网络摄像机密码同步异常
0x54	萤石设备掉线
0x57	配件板异常
0x400	设备开机
0x401	设备关机
0x402	看门狗恢复
0x403	电量低
0x404	电池电压恢复
0x405	交流电断开
0x406	交流电恢复
0x407	网络恢复连接
0x408	闪存读写异常
0x409	读卡器掉线
0x40a	读卡器掉线恢复
0x4000	子板 IP 地址冲突
0x4001	子板掉线

值	描述
0x4002	风扇异常
0x4003	背面板温度异常
0x4004	SD 卡不健康
0x4005	SD 卡损坏

操作日志

值	描述
0x41	开机
0x42	关机
0x43	异常关机
0x44	重启设备（本地）
0x50	登录（本地）
0x51	登出（本地）
0x52	配置（本地）
0x53	按文件回放或下载（本地）
0x54	按时间回放或下载（本地）
0x55	开始录像（本地）
0x56	停止录像（本地）
0x57	云台控制（本地）
0x58	预览（本地，保留）
0x59	编辑时间（本地，保留）
0x5a	升级（本地）
0x5b	视频文件备份（本地）
0x5c	硬盘初始化（本地）
0x5d	导出本地配置文件
0x5e	导入本地配置文件

值	描述
0x5f	文件备份 (本地)
0x60	锁定视频文件 (本地)
0x61	解锁视频文件 (本地)
0x62	手动清除和触发报警 (本地)
0x63	添加网络摄像机 (本地)
0x64	删除网络摄像机 (本地)
0x65	设置网络摄像机 (本地)
0x66	开始备份 (本地)
0x67	停止备份 (本地)
0x68	本地备份开始时间
0x69	本地备份结束时间
0x6a	添加网络硬盘 (本地)
0x6b	删除 NAS (本地)
0x6c	设置 NAS (本地)
0x70	登录 (远程)
0x71	登出 (远程)
0x72	开始录像 (远程)
0x73	停止录像 (远程)
0x74	开始传输
0x75	停止传输
0x76	获取参数 (远程)
0x77	远程配置
0x78	获取状态 (远程)
0x79	布防 (远程)
0x7a	撤防 (远程)
0x7b	重启 (远程)

值	描述
0x7c	开始语音对讲
0x7d	停止语音对讲
0x7e	远程升级
0x7f	按文件回放 (远程)
0x80	按时间回放 (远程)
0x81	云台控制 (远程)
0x82	格式化硬盘 (远程)
0x83	关机 (远程)
0x84	锁定文件 (远程)
0x85	解锁文件 (远程)
0x86	导出配置文件 (远程)
0x87	导入配置文件 (远程)
0x88	导出视频文件 (远程)
0x89	手动清除和触发报警 (远程)
0x8a	添加网络摄像机 (远程)
0x8b	删除网络摄像机 (远程)
0x8c	设置网络摄像机 (远程)
0x8d	重启智能库
0x8e	添加 NAS (远程)
0x8f	删除 NAS (远程)
0x90	设置 NAS (远程)
0x91	开始刻录 (本地)
0x92	停止刻录 (本地)
0x93	开始刻录 (远程)
0x94	开始刻录 (远程)
0x95	图片备份 (本地)

值	描述
0x96	图片备份（远程）
0x97	庭审事件恢复（本地）
0x98	庭审事件恢复（远程）
0x99	导入文件（本地）
0x9a	删除异常或不存在的硬盘
0x9b	加载硬盘（远程）
0x9c	卸载硬盘（远程）
0x9d	锁定（本地）
0x9e	解锁（本地）
0x9f	删除庭审文件（本地）
0xd0	旁路（远程）
0xd1	旁路恢复（远程）
0xd2	设置报警输入参数（远程）
0xd3	获取报警输入参数（远程）
0xd4	设置报警输出参数（远程）
0xd5	获取报警输出参数（远程）
0xd6	手动开启报警输出（远程）
0xd7	手动关闭报警输出（远程）
0xd8	开启或禁用报警主机的 RS-485 串口（远程）
0xd9	导出数据库记录
0xda	导入数据库记录
0xdb	级联切换
0xdc	级联云台控制
0x101	配置自动重建（本地）
0x102	配置热备（本地）
0x103	创建阵列（本地）

值	描述
0x104	删除阵列 (本地)
0x105	迁移阵列 (本地)
0x106	手动重建阵列 (本地)
0x107	一键配置 (本地)
0x108	创建虚拟磁盘 (本地)
0x109	删除虚拟磁盘 (本地)
0x10a	修复虚拟磁盘 (本地)
0x10b	扩展虚拟磁盘 (本地)
0x10c	升级磁盘阵列 (本地)
0x10d	安全拔出磁盘阵列 (本地)
0x111	配置自动重建 (远程)
0x112	配置热备 (远程)
0x113	创建阵列 (远程)
0x114	删除阵列 (远程)
0x115	迁移阵列 (远程)
0x116	手动重建阵列 (远程)
0x117	一键配置 (远程)
0x118	创建虚拟磁盘 (远程)
0x119	删除虚拟磁盘 (远程)
0x11a	修复虚拟磁盘 (远程)
0x11b	扩展虚拟磁盘 (远程)
0x11c	升级磁盘阵列 (远程)
0x11d	安全拔出磁盘阵列 (远程)
0x121	开始抓拍 (本地)
0x122	停止抓拍 (本地)
0x125	设置 SNMP (本地)

值	描述
0x126	标签操作（本地）
0x131	开始抓拍（远程）
0x132	停止抓拍（远程）
0x135	设置 SNMP（远程）
0x136	标签操作（远程）
0x140	切换输出口（本地）
0x141	配置编码性能
0x142	N+1 热备操作（本地）
0x143	N+1 热备操作（远程）
0x144	导出网络摄像机配置文件（本地）
0x145	导入网络摄像机配置文件（本地）
0x146	升级网络摄像机（本地）
0x147	导出网络摄像机配置文件（远程）
0x148	导入网络摄像机配置文件（远程）
0x149	升级网络摄像机（远程）
0x201	设置多屏控制器的主屏
0x202	设置多屏控制器的子屏
0x203	关闭多屏控制器的主屏
0x204	关闭多屏控制器的子屏
0x251	修改输入源
0x252	修改输出通道
0x253	修改虚拟 LED
0x254	修改 LOGO
0x255	设置场景
0x256	显示操作
0x257	获取虚拟 LED
0x258	获取场景

值	描述
0x259	场景控制
0x260	获取所有可用窗口信息
0x261	获取单个窗口信息
0x262	窗口控制
0x263	获取场景列表
0x264	场景控制
0x265	设置单个场景
0x266	获取输入源列表
0x267	获取预案列表
0x268	编辑预案
0x269	控制预案
0x270	屏幕控制
0x271	添加信号源
0x272	编辑信号源
0x273	设置解码板参数
0x274	获取解码板参数
0x275	获取设备信息
0x276	上传背景图片
0x277	设置密码
0x278	添加场景
0x279	删除场景
0x280	删除信号源
0x281	添加预案
0x282	删除预案
0x283	获取外部矩阵参数
0x284	设置外部矩阵参数
0x285	获取用户参数
0x286	设置用户参数

值	描述
0x287	获取电视墙连接参数
0x288	设置电视墙连接参数
0x289	获取电视墙场景信息
0x28a	设置电视墙场景信息
0x28b	获取当前场景信息
0x28c	场景切换
0x300	加载硬盘（本地）
0x301	删除异常或不存在的硬盘（本地）
0x400	开门（远程）
0x401	关门（远程）
0x402	门常开（远程）
0x403	门常闭（远程）
0x404	手动校时（远程）
0x405	自动 NTP 校时
0x406	清除卡号（远程）
0x407	恢复出厂设置（远程）
0x408	防区布防
0x409	防区撤防
0x40a	恢复出厂设置（本地）
0x1001	设置触发模式
0x1002	获取触发模式
0x1003	设置报警输出参数
0x1004	获取报警输出参数
0x1005	获取推荐触发模式
0x1006	获取状态监测参数
0x1007	设置状态监测参数
0x1008	获取状态监测参数

值	描述
0x1009	获取视频电警模式
0x100a	设置视频电警模式
0x2001	添加车辆信息（本地）
0x2002	编辑车辆信息（本地）
0x2003	删除车辆信息（本地）
0x2004	搜索车辆信息（本地）
0x2005	添加布控信息（本地）
0x2006	编辑布控信息（本地）
0x2007	删除布控信息（本地）
0x2008	搜索布控信息（本地）
0x2009	搜索正常通行信息（本地）
0x200a	搜索异常通行信息（本地）
0x200b	搜索正常通行信息（本地）
0x200c	预览图片（本地）
0x200d	设置出入口参数（本地）
0x200e	获取出入口参数（本地）
0x200f	设置数据上传参数（本地）
0x2010	获取数据上传参数（本地）
0x2011	打开或关闭道闸
0x2012	添加外设信息（本地）
0x2013	编辑外设信息（本地）
0x2014	删除外设信息（本地）
0x2015	搜索外设信息（本地）
0x2016	添加停车规则（本地）
0x2017	编辑停车规则（本地）
0x2018	删除停车规则（本地）

值	描述
0x2019	搜索停车规则（本地）
0x2020	正常通过数据统计（本地）
0x2021	导出正常通行统计报告（本地）
0x2022	异常通行数据统计（本地）
0x2023	导出异常通行统计报告（本地）
0x2024	行人通行数据统计
0x2025	导出行人通行统计报告（本地）
0x2026	搜索车辆过路费信息（本地）
0x2027	车辆过路费统计（本地）
0x2028	导出车辆过路费统计报告（本地）
0x2029	搜索排班信息（本地）
0x2030	搜索卡片信息（本地）
0x2031	添加折扣规则（本地）
0x2032	编辑折扣规则（本地）
0x2033	删除折扣规则（本地）
0x2034	搜索折扣规则（本地）
0x2035	获取出入口控制机离线检测参数（本地）
0x2036	设置出入口控制机离线检测参数（本地）
0x2037	设置出入口控制机下发的卡片信息（本地）
0x2038	清除出入口控制机下发的卡片信息（本地）
0x2101	打开或关闭闸机（远程）
0x2102	设置出入口参数（远程）
0x2103	获取出入口参数（远程）
0x2104	设置数据上传参数（远程）
0x2105	获取数据上传参数（远程）
0x2106	获取终端基本信息（远程）

值	描述
0x2107	获取字符叠加参数（远程）
0x2108	设置字符叠加参数（远程）
0x2109	获取路口信息（远程）
0x210a	建立数据同步服务器（远程）
0x210b	获取出入口终端工作状态（远程）
0x210c	获取出入口终端通道状态（远程）
0x210d	添加外设信息（远程）
0x210e	编辑外设信息（远程）
0x210f	获取出入口控制机离线检测参数（远程）
0x2110	设置出入口控制机离线检测参数（远程）
0x2111	设置出入口控制机下发的卡片信息（远程）
0x2112	清除出入口控制机下发的卡片信息（远程）
0x2115	开启车位指示灯控制（远程）
0x2116	禁用车位指示灯控制（远程）
0x2117	设置音量
0x2118	设置录像音量
0x2119	智能配置
0x211a	设置磁盘阵列速度（本地）
0x211b	设置磁盘阵列速度（远程）
0x211c	添加存储池（远程）
0x211d	删除存储池（远程）
0x2120	删除图片（远程）
0x2121	删除录像（远程）
0x2123	启用云存储（远程）
0x2124	禁用云存储（远程）
0x2125	编辑云存储池参数（远程）

值	描述
0x2126	编辑云存储池容量（远程）
0x2200	创建或编辑视图库空间（远程）
0x2201	删除视图库文件（远程）
0x2202	下载视图库文件（远程）
0x2203	上传视图库文件（远程）
0x2204	创建或编辑视图库空间（本地）
0x2501	MCU 会议配置
0x2502	MCU 终端配置
0x2503	MCU 分组配置
0x2504	MCU 会议控制
0x2505	MCU 终端控制

B.5.1 报警主机日志类型

报警日志

日志类型	描述
0x01	短路报警
0x02	短路报警
0x03	报警复位
0x04	报警恢复
0x05	连续三次密码不正确
0x06	卡号无效
0x07	键盘防拆报警
0x08	键盘防拆恢复
0x09	设备防拆报警

日志类型	描述
0x0a	设备防拆恢复
0x0b	模拟量低于报警限 1
0x0c	模拟量低于报警限 2
0x0d	模拟量低于报警限 3
0x0e	模拟量低于报警限 4
0x0f	模拟量高于报警限 1
0x10	模拟量高于报警限 2
0x11	模拟量高于报警限 3
0x12	模拟量高于报警限 4
0x13	紧急按钮按下
0x14	紧急按钮复位
0x15	软防区匪警
0x16	软防区火警
0x17	软防区紧急报警
0x18	移动侦测报警开始
0x19	移动侦测报警结束
0x1a	设备阻塞
0x1b	设备恢复
0x1c	UPS 报警
0x1d	电量表报警
0x1e	供电开关报警
0x1f	气体检测报警
0x20	变电器温度报警

日志类型	描述
0x21	温湿度传感器报警
0x22	UPS 恢复
0x23	电量表恢复
0x24	供电开关恢复
0x25	气体检测恢复
0x26	变电器温度恢复
0x27	温湿度传感器恢复
0x28	温湿度传感器恢复
0x29	水位传感器恢复
0x2a	扬尘噪声传感器报警
0x2b	扬尘噪声传感器恢复
0x2c	环境采集仪报警
0x2d	环境采集仪恢复
0x2e	探测器防拆报警
0x2f	探测器恢复
0x30	紧急报警
0x31	紧急报警复位

异常日志

日志类型	描述
0x01	开机
0x02	关机
0x03	WDT 复位
0x04	电池电压低

日志类型	描述
0x05	交流电断电
0x06	交流电恢复
0x07	RTC 实时时钟异常
0x08	网络连接断开
0x09	网络连接恢复
0x0a	电话线连接断开
0x0b	电话线链接恢复
0x0c	扩展总线连接断开
0x0d	扩展总线连接恢复
0x0e	键盘总线连接断开
0x0f	键盘总线连接恢复
0x10	模拟传感器故障
0x11	模拟传感器恢复
0x12	RS-485 串口连接断开
0x13	RS-485 串口连接恢复
0x14	电池电压恢复
0x15	有线网络异常
0x16	有线网络恢复
0x17	GPRS 异常
0x18	GPRS 恢复
0x19	3G 网络异常
0x1a	3G 网络恢复
0x1b	SIM 卡异常

日志类型	描述
0x1c	SIM 卡恢复
0x1d	视频丢失
0x1e	非法登录
0x1f	硬盘满
0x20	硬盘错误
0x21	调制解调器掉线
0x22	IP 地址冲突
0x23	网络连接断开
0x24	录像异常
0x25	视频输入异常（仅适用于模拟通道）
0x26	远程格式化硬盘失败
0x27	USB 通信故障
0x28	USB 通信恢复
0x29	打印机故障
0x2a	打印机恢复
0x2b	子板通信错误
0x2c	网络摄像机掉线
0x2d	网络摄像机 IP 地址冲突
0x2e	视频制式不匹配

操作日志

日志类型	描述
0x01	普通布防
0x02	普通撤防
0x03	旁路
0x04	挟持
0x05	本地重启
0x06	远程重启
0x07	本地升级
0x08	远程升级
0x09	恢复出厂设置
0x0a	远程报警输出控制
0x0b	门禁：打开
0x0c	门禁关闭
0x0d	警号开启
0x0e	警号关闭
0x0f	防区设置
0x10	报警输出设置
0x11	传感器设置
0x12	RS-485 串口设置
0x13	拨号设置
0x14	添加管理员用户
0x15	编辑管理员用户
0x16	删除管理员用户

日志类型	描述
0x17	添加 DVR/NVR 操作用户
0x18	修改 DVR/NVR 操作用户
0x19	删除 DVR/NVR 操作用户
0x1a	添加摄像机操作用户
0x1b	修改摄像机操作用户的密码
0x1c	删除摄像机操作用户的密码
0x1d	添加键盘/读卡器用户
0x1e	删除键盘/读卡器用户
0x1f	远程登录
0x20	远程注销
0x21	远程布防
0x22	远程撤防
0x23	编辑报警主机参数
0x24	旁路恢复
0x25	启用输出
0x26	禁用输出
0x27	编辑子系统参数
0x28	组旁路
0x29	组旁路恢复
0x2a	编辑 GPRS 参数
0x2b	编辑网络上传参数
0x2c	编辑上传模式
0x2d	编辑门禁参数

日志类型	描述
0x2e	远程：开始录像
0x2f	远程：停止录像
0x30	传输开始
0x31	传输结束
0x32	语音对讲开始
0x33	语音对讲结束
0x34	远程：按文件回放或下载
0x35	远程：按时间回放
0x36	远程：云台控制
0x37	远程：锁定文件
0x38	远程：解锁文件
0x39	远程：格式化硬盘
0x3a	远程：导出配置文件
0x3b	远程：导入配置文件
0x3c	远程：导出文件
0x3d	在家布防
0x3e	即时布防
0x3f	自动布防
0x40	自动撤防
0x41	钥匙防区布防
0x42	钥匙防区撤防
0x43	清除报警
0x44	编辑系统故障参数

日志类型	描述
0x45	编辑报警输出参数
0x46	搜索外设模块
0x47	重新注册外设模块
0x48	关闭键盘报警提示音
0x49	编辑移动网络参数
0x4a	编辑打印机参数
0x4b	格式化 SD 卡
0x4c	升级子板
0x4d	布防或撤防计划配置
0x4e	SMS 布防
0x4f	手机在家布防
0x50	手机即时布防
0x51	手机撤防
0x52	清除手机报警
0x53	白名单设置
0x54	按计划启用或禁用触发配置
0x55	抓拍设置
0x56	防区防拆设置
0x57	远程：解锁
0x58	远程：锁定
0x59	本地：解锁
0x5a	本地：锁定
0x5b	远程：打开警灯

日志类型	描述
0x5c	远程：关闭警灯
0xf0	本地：激活设备
0xf1	远程：激活设备
0xf2	本地：恢复出厂设置
0xf3	远程：恢复出厂设置

事件日志

日志类型	描述
0x01	B 码同步
0x02	SDK 同步
0x03	按计划校时
0x04	插入子板
0x05	拔除子板
0x06	自动布防
0x07	自动撤防
0x08	按计划激活触发器
0x09	按计划反激活触发器
0x0a	自动布防失败
0x0b	自动撤防失败
0x0c	激活触发器失败
0x0d	反激活触发器失败
0x0e	强制布防

B.5.2 日志参数配置

允许 HCISUPSDK 在运行时，进行日志参数的配置。

HCISUPSDK_Log_Switch.xml 文件的具体内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<SdkLocal><!--必选项，SDK 本地配置参数，最外层节点-->
  <CMSLog><!--可选项，HCISUPCMS 模块的日志配置参数-->
    <logLevel>3</logLevel><!--必选项，xs:integer，日志级别，1-ERROR，2-DEBUG，3-INFO-->
    <logDirectory>./CMSLog/</logDirectory><!--必选项，xs:string，日志文件输出目录，可以
    以为相对路径，也可以为绝对路径，必须以“/”结尾-->
    <autoDelete>true</autoDelete><!--必选项，xs:boolean，日志文件是否自动删除，当设置为
    true 时，模块生成的日志文件不会超过 10 个，并进行日志的循环覆盖（第一个日志文件不参与循环覆盖），
    当设置为 false 时，日志数量没有上限-->
  </CMSLog>
  <StreamLog><!--可选项，HCISUPStream 模块的日志配置参数-->
    <logLevel>3</logLevel><!--必选项，xs:integer，日志级别，1-ERROR，2-DEBUG，3-INFO-->
  </StreamLog>
  <AlarmLog><!--可选项，HCISUPAlarm 模块的日志配置参数-->
    <logLevel>3</logLevel><!--必选项，xs:integer，日志级别，1-ERROR，2-DEBUG，3-INFO-->
  </AlarmLog>
  <SSLog><!--可选项，HCISUPSS 模块的日志配置参数-->
    <logLevel>3</logLevel><!--必选项，xs:integer，日志级别，1-ERROR，2-DEBUG，3-INFO-->
  </SSLog>
</SdkLocal>
```


备注

- HCISUPSDK 默认成果物中不含有 HCISUPSDK_Log_Switch.xml。
- 当运行目录中不含有 HCISUPSDK_Log_Switch.xml 配置文件时，模块的日志行为以 SetLogToFile 接口（NET_ECMS SetLogToFile、NET_ESTREAM SetLogToFile、NET_EALARM SetLogToFile、NET_ESS SetLogToFile）设置的日志参数为准。
- 当运行目录中含有 HCISUPSDK_Log_Switch.xml 配置文件时，但 HCISUPSDK_Log_Switch.xml 配置文件 XML 格式不正确，导致 HCISUPSDK 不能正常解析时，模块的日志行为以调用 SetLogToFile 接口（NET_ECMS SetLogToFile、NET_ESTREAM SetLogToFile、NET_EALARM SetLogToFile、NET_ESS SetLogToFile）设置的日志参数为准。
- 当运行目录中含有 HCISUPSDK_Log_Switch.xml 配置文件时，但 HCISUPSDK_Log_Switch.xml 配置文件正确，以配置文件中的配置为准。若配置文件中不含有某模块的配置（例如没有 <SSLog>），则该模块（例如 HCISUPSS 模块）的日志行为以调用 SetLogToFile 接口设置的日志参数为准。
- 当运行目录中含有 HCISUPSDK_Log_Switch.xml 配置文件时，但 HCISUPSDK_Log_Switch.xml 格式正确，但某模块的参数错误时，该模块的日志行为以接口为准，其他正确模块以配置文件配置为准。
- 当运行目录中含有 HCISUPSDK_Log_Switch.xml 配置文件时，但 HCISUPSDK_Log_Switch.xml 格式正确且参数也正确。但之后又修改了配置文件，把配置文件修改坏。则模块的日志行为，以上一次正确的配置（配置文件或接口，但配置文件优先）为准。
- 当通过日志配置文件修改了原因的日志目录时，原有的日志目录及其日志文件不做变动，新的日志打印到新的目录下，该行为等同于调用日志设置接口来调整日志目录。

B.6 支持的网络摄像机

产品类型及系列	产品型号	加密索引	RS-485
50 系列枪机	DS-2CD5046FWD(-、-AP)	DS-2CD5046FWD	支持
	DS-2CD5065FWD(-、-AP)	DS-2CD5065FWD	支持
	DS-2CD5085FWD	DS-2CD5085FWD	支持
	DS-2CD5027FWD-A(B)	DS-2CD5027FWD(B)	支持
	DS-2CD5046F-A(B)	DS-2CD5046F(B)	支持
	DS-2CD5065F(B)	DS-2CD5065F(B)	支持
	DS-2CD5086F(B)	DS-2CD5086F(B)	支持

ISUPSDK（通用） 开发指南

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD5046F-A(C)	DS-2CD5046F(C)	支持
	DS-2CD5065F-A(C)	DS-2CD5065F(C)	支持
	DS-2CD5085F(C)	DS-2CD5085F(C)	支持
51 系列室内半球	DS-2CD5166FWD-IZ(2.8~12mm)	DS-2CD5166FWD-IZ	支持
	DS-2CD5186FWD-IZ(2.8~12mm)	DS-2CD5186FWD-IZ	支持
5A 系列筒机	DS-2CD5A27FWD-IZHS(2.8-12mm)(B)	DS-2CD5A27FWD(B)	支持
	DS-2CD5A27FWD-IZHS(8-32mm)(B)	DS-2CD5A27FWD(B)	支持
	DS-2CD5A46F-IZHS(2.8-12mm)(B)	DS-2CD5A46F(B)	支持
	DS-2CD5A65F-IZHS(2.8-12mm)(B)	DS-2CD5A65F(B)	支持
	DS-2CD5A86F-IZHS(2.8-16mm)(B)	DS-2CD5A86F(B)	支持
	DS-2CD5A46FWD(-IZHS、-IZS、-IZ)	DS-2CD5A46FWD	支持
	DS-2CD5A52FWD(-IZHS、-IZS、-IZ)	DS-2CD5A52FWD	支持
	DS-2CD5A65FWD(-IZHS、-IZS、-IZ)	DS-2CD5A65FWD	支持
	DS-2CD5A85FWD(-IZHS、-IZS、-IZ)	DS-2CD5A85FWD	支持
	DS-2CD5A46F-IZHS(C)	DS-2CD5A46F(C)	支持
	DS-2CD5A52F-IZHS(C)	DS-2CD5A52F(C)	支持
	DS-2CD5A65F-IZHS(C)	DS-2CD5A65F(C)	支持
	DS-2CD5A85F-IZHS(C)	DS-2CD5A85F(C)	支持
	70 系列枪机	DS-2CD7027EWD	DS-2CD7027EWD
DS-2CD7047EWD		DS-2CD7047EWD	支持
DS-2CD7067EWD		DS-2CD7067EWD	支持
DS-2CD70C7EWD-A(国内标配)(白)		DS-2CD70C7EWD	支持

ISUPSDK (通用) 开发指南

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD70C7EWD(国内标配)(白)	DS-2CD70C7EWD	支持
	DS-2CD7087EWD	DS-2CD7087EWD	支持
71 系列半球	DS-2CD7127EWD-IZ(2.8-12mm)(8-32mm)	DS-2CD7127EWD	支持
	DS-2CD7147EWD-IZ(2.8-12mm)	DS-2CD7147EWD	支持
	DS-2CD7167EWD-IZ(2.8-12mm)	DS-2CD7167EWD	支持
	DS-2CD7187EWD-IZ(2.8-12mm)	DS-2CD7187EWD	支持
	DS-2CD7147EWD-IZ(8-32mm)	DS-2CD7147EWD	支持
	DS-2CD7167EWD-IZ(8-32mm)	DS-2CD7167EWD	支持
	DS-2CD71C7EWD-IZ(2.8-12mm)(国内标配)(白)	DS-2CD71C7EWD	支持
	DS-2CD7187EWD-IZ(8-32mm)	DS-2CD7187EWD	支持
7A 系列筒机	DS-2CD7A27EWD/F-(I)(L)Z(S)(2.8-12mm)(8~32mm)	DS-2CD7A27EWD	支持
	DS-2CD7A47EWD-(I)(L)Z(S)(2.8-12mm)	DS-2CD7A47EWD	支持
	DS-2CD7A67EWD-(I)(L)Z(S)(2.8-12mm)	DS-2CD7A67EWD	支持
	DS-2CD7A87EWD-(I)(L)Z(S)(2.8-12mm)	DS-2CD7A87EWD	支持
	DS-2CD7AC7EWD-XZ(S)(2.8-12mm)(国内标配)(白)	DS-2CD7AC7EWD	支持
	DS-2CD7AC7EWD-XZ(S)/ZJ(2.8-12mm)(国内标配)(白)	DS-2CD7AC7EWD	支持
	DS-2CD7A27EWD-XZ(S)(2.8-12mm)(8~32mm)	DS-2CD7A27EWD-X	支持

ISUPSDK（通用） 开发指南

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD7A47EWD-XZ(S)(2.8-12mm) (8~32mm)	DS-2CD7A47EWD-X	支持
	DS-2CD7A67EWD-XZ(S)(2.8-12mm) (8~32mm)	DS-2CD7A67EWD-X	支持
	DS-2CD7A87EWD-XZ(S)(2.8-12mm) (8~32mm)	DS-2CD7A87EWD-X	支持
7D 系列半球	DS-2CD7D27DWD-IZS(W)(4mm-6mm) (6mm-9mm)	DS-2CD7D27DWD	不支持
	DS-2CD7D27DWD-IZS(4-6mm)	DS-2CD7D27DWD	不支持
	DS-2CD7D47DWD-IZS(6-9mm)	DS-2CD7D47DWD	不支持
	DS-2CD7D47DWD-IZS(W)(4mm-6mm) (6mm-9mm)	DS-2CD7D47DWD	不支持
7T 系列筒机	DS-2CD7T27DWD-IZS(GL)(W)(4mm-6mm) (6mm-9mm)	DS-2CD7T27DWD	不支持
	DS-2CD7T27DWD-IZSW(6-9mm)(B)	DS-2CD7T27DWD-B	不支持
	DS-2CD7T27DWD-IZSW(4-6mm)(B)	DS-2CD7T27DWD-B	不支持
	DS-2CD7T47DWD-IZSW(6-9mm)(B)	DS-2CD7T47DWD-B	不支持
	DS-2CD7T47DWD-IZSW(4-6mm)(B)	DS-2CD7T47DWD-B	不支持
	DS-2CD7T27DWD-IZS(6-9mm)(B)	DS-2CD7T27DWD-B	不支持
	DS-2CD7T27DWD-IZS(4-6mm)(B)	DS-2CD7T27DWD-B	不支持
	DS-2CD7T47DWD-IZS(6-9mm)(B)	DS-2CD7T47DWD-B	不支持
	DS-2CD7T47DWD-IZS(4-6mm)(B)	DS-2CD7T47DWD-B	不支持
	DS-2CD7T27DWD-IZS/POE(6-9mm)(B)	DS-2CD7T27DWD-B	不支持
	DS-2CD7T27DWD-IZS/POE(4-6mm)(B)	DS-2CD7T27DWD-B	不支持

ISUPSDK（通用） 开发指南

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD7T47DWD-IZS/POE (6-9mm)(B)	DS-2CD7T47DWD-B	不支持
	DS-2CD7T47DWD-IZS/POE (4-6mm)(B)	DS-2CD7T47DWD-B	不支持
	DS-2CD7T47DWD-IZS(GL)(W)(4mm-6mm)(6mm-9mm)	DS-2CD7T47DWD	不支持
70 系列枪机 (H3+MA)	DS-2CD7027FWD/V-A(/NMFC)(/DN)	DS-2CD7027FWD/V	支持
	DS-2CD7047F/V-A(/NMFC)(/DN)	DS-2CD7047F/V	支持
	DS-2CD7067F/V-A(/NMFC)(/DN)	DS-2CD7067F/V	支持
	DS-2CD7087F/V(/NMFC)(/DN)	DS-2CD7087F/V	支持
	DS-2CD7026FWD-A(/NMFC)(/DN)	DS-2CD7026FWD	支持
	DS-2CD7046F-A(/NMFC)(/DN)	DS-2CD7046F	支持
	DS-2CD7066F-A(/NMFC)(/DN)	DS-2CD7066F	支持
	DS-2CD7086F(/NMFC)(/DN)	DS-2CD7086F	支持
	DS-2CD7046FWD(-、-A)(/NMFC)(/DN)	DS-2CD7046FWD	支持
	DS-2CD7066FWD(-、-A)(/NMFC)(/DN)	DS-2CD7066FWD	支持
	DS-2CD7086FWD(/NMFC)(/DN)	DS-2CD7086FWD	支持
	DS-2CD7047FWD/V(/NMFC)(/DN)	DS-2CD7047FWD/V	支持
	DS-2CD7067FWD/V(/NMFC)(/DN)	DS-2CD7067FWD/V	支持
	DS-2CD7087FWD/V(/NMFC)(/DN)	DS-2CD7087FWD/V	支持
	DS-2CD7027FWD/F-A(/NMFC)(/DN)	DS-2CD7027FWD/F	支持
	DS-2CD7027FWD-A(/NMFC)(/DN)	DS-2CD7027FWD	支持
	DS-2CD7047F/F-A(/NMFC)(/DN)	DS-2CD7047F/F	支持
	DS-2CD7047FWD-A(/NMFC)(/DN)	DS-2CD7047FWD	支持
	DS-2CD7067F/F-A(/NMFC)(/DN)	DS-2CD7067F/F	支持

ISUPSDK（通用） 开发指南

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD7067FWD-A(/NMFC)(/DN)	DS-2CD7067FWD	支持
	DS-2CD7087F/F-A(/NMFC)(/DN)	DS-2CD7087F/F	支持
	DS-2CD7087FWD-A(/NMFC)(/DN)	DS-2CD7087FWD	支持
	DS-2CD7126FWD-IZ(2.8-12mm)(8-32mm)	DS-2CD7126FWD	支持
	DS-2CD7127FWD/F-IZ(2.8-12mm) (8-32mm)	DS-2CD7127FWD	支持
	DS-2CD7127FWD/H-IZ(2.8-12mm)	DS-2CD7127FWD/H	支持
	DS-2CD7147FWD-IZ(2.8-12mm)	DS-2CD7147FWD	支持
	DS-2CD7167FWD-IZ(2.8-12mm)	DS-2CD7167FWD	支持
	DS-2CD7187FWD-IZ(2.8-12mm)	DS-2CD7187FWD	支持
	DS-2CD7147FWD-IZ(8-32mm)	DS-2CD7147FWD	支持
	DS-2CD7167FWD-IZ(8-32mm)	DS-2CD7167FWD	支持
	DS-2CD7187FWD-IZ(8-32mm)	DS-2CD7187FWD	支持
7A 系列筒机 (H3+MA)	DS-2CD7A27FWD/V-(I)(L)ZS(2.8-12mm)	DS-2CD7A27FWD/V	支持
	DS-2CD7A27FWD/V-(I)(L)ZS(8-32mm)	DS-2CD7A27FWD/V	支持
	DS-2CD7A47F/V-(I)(L)ZS(2.8-12mm)	DS-2CD7A47F/V	支持
	DS-2CD7A67F/V-(I)(L)ZS(2.8-12mm)	DS-2CD7A67F/V	支持
	DS-2CD7A87F/V-(I)(L)ZS(2.8-16mm)	DS-2CD7A87F/V	支持
	DS-2CD7A26FWD-(I)(L)ZS(2.8-12mm)	DS-2CD7A26FWD	支持
	DS-2CD7A26FWD-(I)(L)ZS(8-32mm)	DS-2CD7A26FWD	支持
	DS-2CD7A46F-(I)(L)ZS(2.8-12mm)	DS-2CD7A46F	支持
	DS-2CD7A66F-(I)(L)ZS(2.8-12mm)	DS-2CD7A66F	支持
	DS-2CD7A86F-(I)(L)ZS(2.8-16mm)	DS-2CD7A86F	支持

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD7A46FWD(-IZS、-IZ)	DS-2CD7A46FWD	支持
	DS-2CD7A66FWD(-IZS、-IZ)	DS-2CD7A66FWD	支持
	DS-2CD7A86FWD(-IZS、-IZ)	DS-2CD7A86FWD	支持
	DS-2CD7A47FWD/V	DS-2CD7A47FWD/V	支持
	DS-2CD7A67FWD/V(-IZS、-IZ)	DS-2CD7A67FWD/V	支持
	DS-2CD7A87FWD/V(-IZS、-IZ)	DS-2CD7A87FWD/V	支持
	DS-2CD7A27FWD/F-(L)(I)ZS(2.8-12mm)	DS-2CD7A27FWD/F	支持
	DS-2CD7A27FWD/F-(L)(I)ZS(8-32mm)	DS-2CD7A27FWD/F	支持
	DS-2CD7A47F/F-IZS(2.8-12mm)	DS-2CD7A47F/F	支持
	DS-2CD7A47FWD/F-IZS(2.8-12mm)(/NMFC)(/DN)	DS-2CD7A47FWD	支持
	DS-2CD7A67F/F-IZS(2.8-12mm)	DS-2CD7A67F/F	支持
	DS-2CD7A67FWD-IZS(2.8-12mm)(/NMFC)(/DN)(8-32mm)	DS-2CD7A67FWD	支持
	DS-2CD7A87F/F-IZS(2.8-16mm)	DS-2CD7287F/F	支持
	DS-2CD7A87FWD-IZS(2.8-16mm)(/NMFC)(/DN)(8-32mm)	DS-2CD7A87FWD	支持
	DS-2CD7A27FWD-XZ(S)(2.8-12mm)(8-32mm)	DS-2CD7A27FWD-X	支持
	DS-2CD7A47FWD-XZ(S)(2.8-12mm)(8-32mm)	DS-2CD7A47FWD-X	支持
	DS-2CD7A67FWD-XZS(2.8-12mm)(8-32mm)	DS-2CD7A67FWD-X	支持

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD7A87FWD-XZS(2.8-12mm)(8-32mm)	DS-2CD7A87FWD-X	支持
72 系列枪机 (H3+MA)	DS-2CD7287F/V(5.7-21mm, 15 - 55mm 手动变焦和电动变焦)	DS-2CD7287F/V	支持
	DS-2CD72C7FWD/V (25mm, 30mm, 50mm 定焦镜头)	DS-2CD72C7FWD/V	支持
	DS-2CD7287F/F	DS-2CD7287F/F	支持
	DS-2CD72C7FWD/F	DS-2CD72C7FWD/F	支持
78 系列筒机 (H3+MA)	DS-2CD7826FWD-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)(8-32mm)	DS-2CD7826FWD	支持
	DS-2CD7827FWD/V-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)(8-32mm)	DS-2CD7827FWD/V	支持
	DS-2CD7827FWD/F-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)(8-32mm)	DS-2CD7827FWD/F	支持
	DS-2CD7846FWD-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)	DS-2CD7846FWD	支持
	DS-2CD7847FWD/V-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)	DS-2CD7847FWD/V	支持
	DS-2CD7847F/F-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)	DS-2CD7847F/F	支持
	DS-2CD7866FWD-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)	DS-2CD7866FWD	支持
	DS-2CD7867FWD/V-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)	DS-2CD7867FWD/V	支持
	DS-2CD7867F/F-(I)(L)Z(S)(/SWG)(/GLG)(2.8-12mm)	DS-2CD7867F/F	支持

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD7886F-(I)(L)Z(S)(/SWG)(/GLG) (2.8-16mm)	DS-2CD7886F	支持
	DS-2CD7887F/V-(I)(L)Z(S)(/SWG)(/GLG) (2.8-16mm)	DS-2CD7887F/V	支持
	DS-2CD7887F/F-(I)(L)Z(S)(/SWG)(/GLG) (2.8-16mm)	DS-2CD7887F/F	支持
护罩一体机 (H3+MA)	DS-2CDx0xx/E-IN(5.7-21mm)(3.8-16mm) (11-40mm)	/	/
	DS-2CDx0xx/E-(I)(L)Z(J)(5.7-21mm) (11-40mm)(15-55mm)	/	/
	DS-GPZ2xA7/VE-(I)(L)SWG(D90)(Q70)	/	/
	DS-GPZ2xA7/FE-(I)(L)SWG(D90)(Q70)	/	/
防爆人脸 (H3+MA)	DS-2XE8427FWD-LZHS(8-32mm ZF2-D) (2.8-12mm ZF11)	DS-2XE8427FWD	支持
	DS-2XE8447FWD-(L)(I)ZHS(8-32mm ZF15)	DS-2XE8447FWD	支持
	DS-2XE8487FWD-(L)(I)ZHS(8-32mm ZF15)	DS-2XE8487FWD	支持

B.7 支持的网络硬盘录像机

I 系列网络硬盘录像机

产品型号
DS-9664N-I16
DS-9664N-I8
DS-9616N-I8(8*8T AI 盘)

产品型号
DS-9632N-I8
DS-9616N-I8
DS-9608N-I8
DS-9616N-I16
DS-9632N-I16
DS-FN64DGX(B)
DS-9664N-I16(16*8T AI 盘)
DS-FN16DGX(B)
DS-9616N-I16(16*8T AI 盘)
DS-FN32DGX(B)
DS-9632N-I16(16*8T AI 盘)
DS-9632N-I8(8*8T AI 盘)
DS-8632N-I8/24P
DS-8616N-I8
DS-FN32CGX(B)
DS-8608N-I8(8*6T)
DS-8616N-I8(8*6T)
DS-8632N-I8
DS-8664N-I8
DS-FN08CGX(B)
DS-8664N-I16
DS-FN08MG

产品型号
DS-8616N-I16
DS-8632N-I8(8*6T)
DS-8608N-I8
DS-8664N-I8/MB-84
DS-FN16CGX(B)
DS-8632N-I16(16*6T)
DS-FN16DG(C)
DS-FN08PG(B)
DS-8632N-I16
DS-FN64CG(C)
DS-FN16MG
DS-FN16PG(B)
DS-8632N-I8/MB-84
DS-FN32DG(C)
DS-8616N-I8/MB-84
DS-8664N-I8(8*6T)
DS-GJZ9N08-F86(B)
DS-FN32MG
DS-8608N-I8/MB-84
DS-8616N-I16(16*6T)
DS-8664N-I16(16*6T)
DS-GJZ9N08-F96(B)

产品型号
DS-FN64CGX(B)
DS-7732N-I4
DS-7708N-I4
DS-7716N-I4/16P
DS-7732N-I4/16P
DS-7708N-I4/8P
DS-7732N-I4/24P
DS-7716N-I4
DS-7608N-I2
DS-7616N-I2
DS-7708N-I4/8P/1G
DS-7616N-I2/16P
DS-7608N-I2/8P
DS-7632N-I2/16P
DS-7632N-I2

B.8 国家和地区代码

车牌识别的算法库支持的国家和地区代码列表如下。

CRIndex（国家和地区代码）	国家和地区
0	算法不支持
1	捷克共和国
2	法国

CRIndex (国家和地区代码)	国家和地区
3	德国
4	西班牙
5	意大利
6	荷兰
7	波兰
8	斯洛伐克
9	白俄罗斯
10	摩尔多瓦
11	俄罗斯
12	乌克兰
13	比利时
14	保加利亚
15	丹麦
16	芬兰
17	英国
18	希腊
19	克罗地亚
20	匈牙利
21	以色列(亚洲)
22	卢森堡
23	马其顿共和国 (2018年改名为北马其顿共和国)
24	挪威
25	葡萄牙
26	罗马尼亚

CRIndex (国家和地区代码)	国家和地区
27	塞尔维亚
28	阿塞拜疆共和国(亚洲)
29	格鲁吉亚(亚洲)
30	哈萨克斯坦(亚洲)
31	立陶宛共和国
32	土库曼斯坦(亚洲)
33	乌兹别克斯坦(亚洲)
34	拉脱维亚
35	爱沙尼亚
36	阿尔巴尼亚
37	奥地利
38	波斯尼亚和黑塞哥维那
39	爱尔兰
40	冰岛
41	梵蒂冈
42	马耳他
43	瑞典
44	瑞士
45	塞浦路斯
46	土耳其
47	斯洛文尼亚
48	黑山共和国
49	科索沃

CRIndex (国家和地区代码)	国家和地区
50	安道尔
51	亚美尼亚(亚洲)
52	摩纳哥
53	列支敦士登
54	圣马力诺
55	保留字段
56	保留字段
57	保留字段
58	保留字段
59	中国
60	巴林
61	韩国
62	黎巴嫩
63	尼泊尔
64	泰国
65	巴基斯坦
66	阿拉伯联合酋长国
67	不丹
68	阿曼
69	朝鲜
70	菲律宾
71	柬埔寨
72	卡塔尔
73	吉尔吉斯斯坦

CRIndex (国家和地区代码)	国家和地区
74	马尔代夫
75	马来西亚
76	蒙古
77	沙特阿拉伯
78	文莱
79	老挝
80	日本
81	土耳其
82	巴勒斯坦国
83	塔吉克斯坦
84	科威特
85	叙利亚
86	印度
87	印度尼西亚
88	阿富汗
89	斯里兰卡
90	伊拉克
91	越南
92	伊朗
93	也门
94	约旦
95	缅甸
96	锡金
97	孟加拉国

CRIndex (国家和地区代码)	国家和地区
98	新加坡
99	东帝汶
100	保留字段
101	保留字段
102	保留字段
103	保留字段
104	埃及
105	利比亚
106	苏丹
107	突尼斯
108	阿尔及利亚
109	摩洛哥
110	埃塞俄比亚
111	厄立特里亚
112	索马里
113	吉布提
114	肯尼亚
115	坦桑尼亚
116	乌干达
117	卢旺达
118	布隆迪
119	塞舌尔
120	乍得
121	中非

CRIndex (国家和地区代码)	国家和地区
122	喀麦隆
123	赤道几内亚
124	加蓬
125	刚果共和国 (即:刚果 (布))
126	刚果民主共和国 (即:刚果 (金))
127	圣多美和普林西比
128	毛里塔尼亚
129	西撒哈拉 (撒拉威)
130	塞内加尔
131	冈比亚
132	马里
133	布基纳法索
134	几内亚
135	几内亚比绍
136	佛得角
137	塞拉利昂
138	利比里亚
139	科特迪瓦
140	加纳
141	多哥
142	贝宁
143	尼日尔
144	赞比亚

CRIndex (国家和地区代码)	国家和地区
145	安哥拉
146	津巴布韦
147	马拉维
148	莫桑比克
149	博茨瓦纳
150	纳米比亚
151	南非
152	斯威士兰
153	莱索托
154	马达加斯加
155	科摩罗
156	毛里求斯
157	尼日利亚
158	南苏丹
159	圣赫勒拿 (英国海外省)
160	马约特 (法国海外省)
161	留尼旺 (法国海外省)
162	加那利群岛 (西班牙海外领土)
163	亚速尔群岛 (北大西洋非洲西海岸 (葡萄牙))
164	马德拉群岛 (北大西洋非洲西海岸 (葡萄牙))
165	保留字段
166	保留字段
167	保留字段

CRIndex (国家和地区代码)	国家和地区
168	保留字段
169	加拿大
170	格陵兰 (丹麦王国的海外自治领土)
171	圣皮埃尔和密克隆 (法国海外领土)
172	美国
173	百慕大 (英国海外领土)
174	墨西哥
175	危地马拉
176	伯利兹
177	萨尔瓦多
178	洪都拉斯
179	尼加拉瓜
180	哥斯达黎加
181	巴拿马
182	巴哈马
183	特克斯群岛和凯科斯群岛 (英国海外领土)
184	古巴
185	牙买加
186	开曼群岛 (英国海外领土)
187	海地
188	多米尼加
189	波多黎各 (美国海外领土)
190	美属维尔京群岛

CRIndex (国家和地区代码)	国家和地区
191	英属维尔京群岛
192	安圭拉 (英国海外领土)
193	安提瓜和巴布达
194	法属圣马丁 (法国海外领土)
195	荷属圣马丁 (荷兰王国的自治国)
196	圣巴泰勒米岛 (法国海外领土)
197	圣基茨和尼维斯
198	蒙特塞拉特 (英国海外领土)
199	瓜德罗普 (法国海外领土)
200	多米尼克
201	马提尼克 (法国海外领土)
202	圣卢西亚
203	圣文森特和格林纳丁斯
204	格林纳达
205	巴巴多斯
206	特立尼达和多巴哥
207	库拉索 (荷兰王国的一个自治国)
208	阿鲁巴 (荷兰王国的一个自治国)
209	荷属安的列斯
210	哥伦比亚
211	委内瑞拉
212	圭亚那
213	苏里南

CRIndex (国家和地区代码)	国家和地区
214	法属圭亚那
215	厄瓜多尔
216	秘鲁
217	玻利维亚
218	巴拉圭
219	智利
220	巴西
221	乌拉圭
222	阿根廷
223	保留字段
224	保留字段
225	保留字段
226	保留字段
227	澳大利亚
228	新西兰
229	巴布亚新几内亚
230	所罗门群岛
231	瓦努阿图
232	新喀里多尼亚 (法国的海外属地)
233	帕劳
234	密克罗尼西亚联邦
235	马绍尔群岛
236	北马里亚纳群岛 (美国的海外属地)

CRIndex（国家和地区代码）	国家和地区
237	关岛（美国的海外属地）
238	瑙鲁
239	基里巴斯
240	斐济群岛
241	汤加
242	图瓦卢
243	瓦利斯和富图纳（法国的海外属地）
244	萨摩亚
245	美属萨摩亚
246	托克劳（新西兰）
247	纽埃（新西兰）
248	库克群岛（新西兰）
249	法属波利尼西亚
250	皮特凯恩群岛（英国的海外属地）
251	夏威夷（美国的海外属地）
252	保留字段
253	保留字段
254	无法识别
255	全部
256	中国台湾
257	中国香港
258	中国澳门

B.9 区域码

车牌识别算法库支持的区域码列表如下。

region（区域码）	区域
ER	俄语区域
EU	欧洲
EUandCIS	欧洲和俄罗斯
ME	中东
other	其他
APAC	亚太区
AFandAM	非洲和美洲
THAandLA	泰国老挝区
HKandMO	港澳区
India	印度
All	所有区域

