# 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,本文档是"按照现状"提供,可能存在瑕疵或错误。本公司不对本文档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文档导致的任何特殊、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	2
	1	. 3. 1 功能模块	2
	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	4
	1.5	注意事项	5
第	2章	设备注册	6
	2. 1	回调函数的注册流程	6
第	3章	预览	10
第	4章	文件查找	19
第	5 章	回放	23
第	6章	语音对讲	33
	6. 1	开启语音对讲	33
	6.2	语音转发	37
	6.3	通过流媒体服务开启语音对讲	38
第	7章	云台控制	47
第	8 章	报警或事件配置	48
第	9章	开启监听服务并接收报警	52
第	10 章	设备维护	57
第	11 章	存储管理	58
	11.1	开启存储服务	59
	11.2	上传文件至存储服务	62
	11 3	从存储服条下裁文件	66

第	12 章	·数配置 70
第	13 章	· 【日参考 75
	13. 1	E册模块 75
	1	1.1 基础接口 75
	1	1.2 配置和控制接口 78
	1	1.3 监听接口 92
	1	1.4 预览接口 94
	1	1.5 回放接口 9分
	1	1.6 语音对讲接口 105
	1	1.7 报文传输接口 10位
	13. 2	· 警监听模块 11
	1	2.1 基础接口 11
	1	2.2 配置和控制接口 120
	1	2.3 监听接口 12
	13.3	ス流模块 125
	1	3.1 基础接口 125
	1	3.2 预览接口 12′
	1	3.3 回放接口 130
	1	3.4 语音对讲接口 135
	13.4	K储管理模块 138
	1	4.1 基础接口 13.
	1	4.2 监听接口 138
	1	4.3 上传和下载客户端接口 140
第	14 章	]调函数 14
	14. 1	SYNC_RESPONSE_CB
	14. 2	EVICE_REGISTER_CB
	14. 3	CMSCallback
	14.4	HomeMsgCallBack

	14.5 EHomeSSMsgCallBack	147
	14.6 EHomeSSRWCallBack	147
	14.7 EHomeSSStorageCallBack	148
	14.8 fExceptionCallBack	149
	14.9 fVoiceDataCallBack	150
	14.10 LongConfigCallBack	151
	14.11 PASSTHROUGHDATACALLBACK	151
	14.12 PLAYBACK_DATA_CB	152
	14.13 PLAYBACK_NEWLINK_CB	153
	14.14 PREVIEW_DATA_CB	154
	14.15 PREVIEW_NEWLINK_CB	154
	14.16 VOICETALK_DATA_CB	155
	14.17 VOICETALK_NEWLINK_CB	155
第	15 章 结构体和枚举	157
	15.1 数据结构体	157
	15. 1. 1 NET_DVR_TYPE_NAME	157
	15. 1. 2 NET_EHOME_ADB_DEBUG	157
	15. 1. 3 NET_EHOME_ALARM_HOST_LOG	158
	15. 1. 4 NET_EHOME_ALARM_HOST_LOG_COND	159
	15.1.5 NET_EHOME_ALARM_INFO	160
	15.1.6 NET_EHOME_ALARM_ISAPI_INFO	161
	15.1.7 NET_EHOME_ALARM_ISAPI_PICDATA	162
	15.1.8 NET_EHOME_ALARM_LISTEN_PARAM	163
	15. 1. 9 NET_EHOME_ALARM_MPDCDATA	164
	15. 1. 10 NET_EHOME_ALARM_MSG	165
	15. 1. 11 NET_EHOME_ALARM_STATUS_UNION	167
	15. 1. 12 NET_EHOME_ALARM_TIME_CFG	168

15. 1. 14 NET_EHOME_ALARMIN_CFG	169
15. 1. 15 NET_EHOME_ALARMIN_COND	170
15. 1. 16 NET_EHOME_ALARMIN_LINKAGE_TYPE	171
15. 1. 17 NET_EHOME_ALARMOUT_CFG	172
15. 1. 18 NET_EHOME_ALARMOUT_STATUS_CFG	172
15. 1. 19 NET_EHOME_ALARMWIRELESSINFO	173
15. 1. 20 NET_EHOME_AMS_ADDRESS	173
15. 1. 21 NET_EHOME_ASYNC_RESP_CB_DATA	174
15. 1. 22 NET_EHOME_BINARY_RECV_DATA	175
15. 1. 23 NET_EHOME_BINARY_SEND_DATA	177
15. 1. 24 NET_EHOME_BLACKLIST_SEVER	180
15. 1. 25 NET_EHOME_CHAN_STATUS_CHANGED	180
15. 1. 26 NET_EHOME_CHAN_TIMING_STATUS_SINGLE	181
15. 1. 27 NET_EHOME_CHAR_EFFECT	182
15. 1. 28 NET_EHOME_CID_INFO	182
15. 1. 29 NET_EHOME_CID_INFO_INTERNAL_EX	184
15. 1. 30 NET_EHOME_CID_PARAM	185
15. 1. 31 NET_EHOME_CMS_LISTEN_PARAM	186
15. 1. 32 NET_EHOME_CMSCB_DATA	186
15. 1. 33 NET_EHOME_COMPONENT_GROUP_INFO	188
15. 1. 34 NET_EHOME_COMPRESSION_CFG	189
15. 1. 35 NET_EHOME_COMPRESSION_COND	191
15. 1. 36 NET_EHOME_CONFIG	191
15. 1. 37 NET_EHOME_DEV_DEFAULT_PARAM	192
15. 1. 38 NET_EHOME_DEV_LOG	193
15. 1. 39 NET_EHOME_DEV_LOG_COND	195
15. 1. 40 NET_EHOME_DEV_REG_INFO	196
15. 1. 41 NET EHOME DEV REG INFO V12	197

15. 1. 42 NET_EHOME_DEV_SESSIONKEY	198
15. 1. 43 NET_EHOME_DEV_SESSIONKEY_EHOME50	198
15. 1. 44 NET_EHOME_DEV_STATUS_CHANGED	199
15. 1. 45 NET_EHOME_DEV_TIMING_STATUS	199
15. 1. 46 NET_EHOME_DEV_WORK_STATUS	200
15. 1. 47 NET_EHOME_DEVICE_CFG	202
15. 1. 48 NET_EHOME_DEVICE_INFO	203
15. 1. 49 NET_EHOME_DISCOVERY_MODE	205
15. 1. 50 NET_EHOME_DISPLAY_PARAM	206
15. 1. 51 NET_EHOME_ERRCODE_RET	208
15. 1. 52 NET_EHOME_ETHERNET	208
15. 1. 53 NET_EHOME_FACESNAP_REPORT	209
15. 1. 54 NET_EHOME_FLOW_COND	211
15. 1. 55 NET_EHOME_FLOW_INFO	212
15. 1. 56 NET_EHOME_GPS_CFG	212
15. 1. 57 NET_EHOME_GPS_INFO	213
15. 1. 58 NET_EHOME_HD_STATUS_CHANGED	215
15. 1. 59 NET_EHOME_HD_TIMING_STATUS_SINGLE	216
15. 1. 60 NET_EHOME_HEATMAP_REPORT	216
15. 1. 61 NET_EHOME_HEATMAP_VALUE	217
15. 1. 62 NET_EHOME_HUMAN_FEATURE	218
15. 1. 63 NET_EHOME_IDENTIFICATION	219
15. 1. 64 NET_EHOME_IMAGE_CFG	219
15. 1. 65 NET_EHOME_INSERT_CHARACTER	220
15. 1. 66 NET_EHOME_INSERT_EFFECT	220
15. 1. 67 NET_EHOME_INSERT_INFO	221
15. 1. 68 NET_EHOME_INSERT_INFO_V20	222
15. 1. 69 NET_EHOME_INSERT_TEXT_INFO	222

15. 1. 70 NET_EHOME_IPADDR	223
15. 1. 71 NET_EHOME_IPADDRESS	224
15. 1. 72 NET_EHOME_IPC_INFO	224
15. 1. 73 NET_EHOME_LINKAGE_ALARMOUT	226
15. 1. 74 NET_EHOME_LINKAGE_PTZ	226
15. 1. 75 NET_EHOME_LISTEN_PREVIEW_CFG	227
15. 1. 76 NET_EHOME_LISTEN_VOICETALK_CFG	228
15. 1. 77 NET_EHOME_LOCAL_ACCESS_SECURITY	228
15. 1. 78 NET_EHOME_LOCAL_GENERAL_CFG	229
15. 1. 79 NET_EHOME_LONG_CFG_INPUT	229
15. 1. 80 NET_EHOME_LONG_CFG_SEND	230
15. 1. 81 NET_EHOME_MAKE_I_FRAME	230
15. 1. 82 NET_EHOME_MANUAL_IOOUT_CTRL	231
15. 1. 83 NET_EHOME_MPDATA	232
15. 1. 84 NET_EHOME_MPGPS	233
15. 1. 85 NET_EHOME_NETWORK_CFG	233
15. 1. 86 NET_EHOME_NEWLINK_CB_MSG	235
15. 1. 87 NET_EHOME_NOTICE_PICURL	235
15. 1. 88 NET_EHOME_NOTIFY_FAIL_INFO	237
15. 1. 89 NET_EHOME_PASSTHROUGH_PARAM	238
15. 1. 90 NET_EHOME_PASSWORD_CFG	238
15. 1. 91 NET_EHOME_PIC_CFG	239
15. 1. 92 NET_EHOME_PIC_FILE	240
15. 1. 93 NET_EHOME_PIC_FILE_COND	241
15. 1. 94 NET_EHOME_PIXEL_ARRAY_SIZE	243
15. 1. 95 NET_EHOME_PLANCTRLPARAM	243
15. 1. 96 NET_EHOME_PLAYBACK_DATA_CB_INFO	244
15. 1. 97 NET_EHOME_PLAYBACK_DATA_CB_PARAM	244

15.1.98 N	NET_EHOME_PLAYBACK_INFO_IN	245
15.1.99 N	NET_EHOME_PLAYBACK_INFO_OUT	247
15. 1. 100	NET_EHOME_PLAYBACK_LISTEN_PARAM	247
15. 1. 101	NET_EHOME_PLAYBACK_NEWLINK_CB_INFO	248
15. 1. 102	NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM	249
15. 1. 103	NET_EHOME_PLAYER_PARAM	249
15. 1. 104	NET_EHOME_POINT	250
15. 1. 105	NET_EHOME_POSITION_INFO	250
15. 1. 106	NET_EHOME_POST_PUBLISH_XML	251
15. 1. 107	NET_EHOME_POST_SCHEDULE	252
15. 1. 108	NET_EHOME_PPPOECFG	253
15. 1. 109	NET_EHOME_PRESET_PARAM	253
15. 1. 110	NET_EHOME_PREVIEW_CB_MSG	254
15. 1. 111	NET_EHOME_PREVIEW_DATA_CB_PARAM	255
15. 1. 112	NET_EHOME_PREVIEWINFO_IN_V11	255
15. 1. 113	NET_EHOME_PREVIEWINFO_OUT	256
15. 1. 114	NET_EHOME_PT_PARAM	257
15. 1. 115	NET_EHOME_PTXML_PARAM	257
15. 1. 116	NET_EHOME_PTZ_PARAM	258
15. 1. 117	NET_EHOME_PUBLISH_SERVERADDR	260
15. 1. 118	NET_EHOME_PUSHPLAYBACK_IN	261
15. 1. 119	NET_EHOME_PUSHPLAYBACK_OUT	261
15. 1. 120	NET_EHOME_PUSHSTREAM_IN	262
15. 1. 121	NET_EHOME_PUSHSTREAM_OUT	262
15. 1. 122	NET_EHOME_PUSHVOICE_IN	263
15. 1. 123	NET_EHOME_PUSHVOICE_OUT	263
15. 1. 124	NET_EHOME_PZIN_PARAM	264
15. 1. 125	NET_EHOME_QUEUE_DATA	264

15. 1. 126	NET_EHOME_QUEUE_DATALIST	265
15. 1. 127	NET_EHOME_QUEUE_INFO	265
15. 1. 128	NET_EHOME_QUEUE_ITEM_DATALIST	266
15. 1. 129	NET_EHOME_REC_FILE	266
15. 1. 130	NET_EHOME_REC_FILE_COND	268
15. 1. 131	NET_EHOME_RECORD_CHAN	269
15. 1. 132	NET_EHOME_RELATE_IPC_INFO	270
15. 1. 133	NET_EHOME_RELEASE_SERVER_INFO	270
15. 1. 134	NET_EHOME_REMOTE_CTRL_PARAM	272
15. 1. 135	NET_EHOME_REPLACE_MATERIAL	273
15. 1. 136	NET_EHOME_SCHEDTIME	274
15. 1. 137	NET_EHOME_SCREEN_SHOT	274
15. 1. 138	NET_EHOME_SCREEN_SHOT_EX	275
15. 1. 139	NET_EHOME_SCREEN_SHOT_RET	275
15. 1. 140	NET_EHOME_SEND_PARAM	276
15. 1. 141	NET_EHOME_SERVER_INFO	276
15. 1. 142	NET_EHOME_SERVER_INFO_V50	278
15. 1. 143	NET_EHOME_SET_REREGISTER_MODE	279
15. 1. 144	NET_EHOME_SHOT_PIC	280
15. 1. 145	NET_EHOME_SS_CLIENT_PARAM	280
15. 1. 146	NET_EHOME_SS_LISTEN_HTTPS_PARAM	281
15. 1. 147	NET_EHOME_SS_LISTEN_PARAM	282
15. 1. 148	NET_EHOME_SS_LOCAL_SDK_PATH	283
15. 1. 149	NET_EHOME_SS_TOMCAT_MSG	284
15. 1. 150	NET_EHOME_STOPPLAYBACK_PARAM	287
15. 1. 151	NET_EHOME_STOPSTREAM_PARAM	287
15. 1. 152	NET_EHOME_STOPVOICETALK_STM_PARAM	288
15. 1. 153	NET EHOME SWITCH DAILY PLAN	288

15. 1. 154	NET_EHOME_SWITCH_PLAN	289
15. 1. 155	NET_EHOME_SWITCH_PLAN_PARAM	289
15. 1. 156	NET_EHOME_SWITCH_WEEKLY_PLAN	290
15. 1. 157	NET_EHOME_SWITCHE_DAY_OF_WEEK_PLAN	290
15. 1. 158	NET_EHOME_SYSCOMPONENT_REDUCED_ADDR	291
15. 1. 159	NET_EHOME_TERM_DEFAULT_GATEWAY	292
15. 1. 160	NET_EHOME_TERM_IP_ADDRESS	292
15. 1. 161	NET_EHOME_TERM_TIME_ZONE	293
15. 1. 162	NET_EHOME_TERMINAL_CONTROL	293
15. 1. 163	NET_EHOME_TERMINAL_CONTROL_V20	294
15. 1. 164	NET_EHOME_TERMINAL_INFO	294
15. 1. 165	NET_EHOME_TERMINAL_NAME	296
15. 1. 166	NET_EHOME_TERMINAL_PROGRESS	296
15. 1. 167	NET_EHOME_TERMINAL_REPORT_INFO	297
15. 1. 168	NET_EHOME_TERMINAL_STATE	298
15. 1. 169	NET_EHOME_TERMINAL_UPG_ST_REPORT	298
15. 1. 170	NET_EHOME_TEXT_MESSAGE	299
15. 1. 171	NET_EHOME_TIME	300
15. 1. 172	NET_EHOME_TIME_ADJUST_PARAM	301
15. 1. 173	NET_EHOME_UPGRADE_CONN_PARAM	301
15. 1. 174	NET_EHOME_VERSION_INFO	301
15. 1. 175	NET_EHOME_VOICE_TALK_IN	302
15. 1. 176	NET_EHOME_VOICE_TALK_OUT	303
15. 1. 177	NET_EHOME_VOICETALK_DATA	303
15. 1. 178	NET_EHOME_VOICETALK_DATA_CB_INFO	304
15. 1. 179	NET_EHOME_VOICETALK_DATA_CB_PARAM	304
15. 1. 180	NET_EHOME_VOICETALK_NEWLINK_CB_INFO	305
15. 1. 181	NET EHOME VOICETALK PARA	305

	15. 1. 182 NET_EHOME_VOLUME_DAILY_PLAN	306
-	15.1.183 NET_EHOME_VOLUME_DAY_OF_WEEK_PLAN	307
-	15. 1. 184 NET_EHOME_VOLUME_PLAN	307
-	15. 1. 185 NET_EHOME_VOLUME_PLAN_PARAM	308
-	15. 1. 186 NET_EHOME_VOLUME_WEEKLY_PLAN	309
-	15. 1. 187 NET_EHOME_WEATHER_INFO	309
-	15. 1. 188 NET_EHOME_WIRELESS_INFO_CFG	310
-	15. 1. 189 NET_EHOME_XML_CFG	310
-	15. 1. 190 NET_EHOME_XML_REMOTE_CTRL_PARAM	311
-	15. 1. 191 NET_EHOME_ZONE	313
15. 2	2 枚举定义	313
-	15.2.1 EN_ALARM_TYPE	313
-	15.2.2 LONG_LINK_MSG	319
-	15. 2. 3 NET_CMS_ENUM_PROXY_TYPE	320
-	15. 2. 4 NET_EHOME_ADDRESS_TYPE	320
-	15. 2. 5 NET_EHOME_ADDRFORMAT_CMD	320
-	15. 2. 6 NET_EHOME_CALLBACK_TYPE	321
-	15. 2. 7 NET_EHOME_CMS_INIT_CFG_TYPE	321
-	15. 2. 8 NET_EHOME_CONTROL_TYPE	322
-	15. 2. 9 NET_EHOME_DAY_OF_WEEK1	323
-	15. 2. 10 NET_EHOME_EALARM_INIT_CFG_TYPE	324
-	15. 2. 11 NET_EHOME_ERROR_CODE	325
-	15. 2. 12 NET_EHOME_ESTREAM_INIT_CFG_TYPE	325
-	15. 2. 13 NET_EHOME_INSERT_STATE	326
-	15. 2. 14 NET_EHOME_IPC_TYPE	327
-	15. 2. 15 NET_EHOME_LOCAL_CFG_TYPE	327
	15. 2. 16 NET_EHOME_PLANCTRL_TYPE	328
	15.2.17 NET EHOME PLAY SCHEDULE TYPE1	329

	15. 2. 18 NET_EHOME_PLAYBACK_OPERATE_MODE	329
	15. 2. 19 NET_EHOME_POS_MODE	330
	15. 2. 20 NET_EHOME_POWER_ONOFF	330
	15. 2. 21 NET_EHOME_PROGRAM_INSERT_TYPE	331
	15. 2. 22 NET_EHOME_REFRESH_TYPE	331
	15. 2. 23 NET_EHOME_REGISTER_TYPE	332
	15. 2. 24 NET_EHOME_SS_CLIENT_TYPE	333
	15. 2. 25 NET_EHOME_SS_INIT_CFG_TYPE	333
	15. 2. 26 NET_EHOME_SS_MSG_TYPE	334
	15. 2. 27 NET_EHOME_STREAM_TYPE	334
	15. 2. 28 NET_EHOME_TERMINAL_PLAY_STATE	335
	15. 2. 29 NET_EHOME_TERMINAL_TYPE	336
	15. 2. 30 NET_EHOME_THIRD_PARTY_DATA_TYPE	336
	15. 2. 31 NET_EHOME_TIME_ZONE	337
	15. 2. 32 NET_EHOME_TRANSMINT_PROTOCAL	338
	15. 2. 33 SEARCH_TYPE	339
第 16	章 命令和报文	340
16	5.1 控制命令报文	340
	16. 1. 1 ADJUSTTIME	340
	16. 1. 2 GETDEVICEWORKSTATUS	341
	16. 1. 3 PZIN	342
	16. 1. 4 REBOOT	343
	16. 1. 5 RESTARTIVSLIB	344
	16. 1. 6 SETDEFENSESTATUS	344
	16. 1. 7 UPDATE	345
16	5.2 配置命令报文	346
	16.2.1 GetAlarmDeploymentTime	346
	16.2.2 GetAlarmLinkageType	347

16.2.3 GetAlarmTriggerCapture	348
16.2.4 GetAlarmTriggerRecord	349
16.2.5 GetCalibrationPara	350
16.2.6 GetDevAbility	351
16.2.7 GetExceptionPara	351
16.2.8 GetHideAlarmArea	352
16.2.9 GetHideAlarmPara	353
16. 2. 10 GetHideArea	354
16.2.11 GetHidePara	355
16.2.12 GetMotionArea	356
16.2.13 GetMotionPara	357
16. 2. 14 GetPassengerLinePara	358
16. 2. 15 GetPassengerOSDPara	359
16. 2. 16 GetPassengerPara	359
16. 2. 17 GetRecordPlanPara	360
16. 2. 18 GetRS232Para	361
16. 2. 19 GetRS485Para	362
16. 2. 20 GetVILostPara	363
16.2.21 SetAlarmDeploymentTime	364
16. 2. 22 SetAlarmLinkageType	365
16. 2. 23 SetAlarmTriggerCapture	366
16. 2. 24 SetAlarmTriggerRecord	367
16.2.25 SetCalibrationPara	368
16. 2. 26 SetExceptionPara	369
16.2.27 SetHideAlarmArea	370
16.2.28 SetHideAlarmPara	370
16. 2. 29 SetHideArea	371
16.2.30 SetHidePara	372

	16.2.31 SetMotionArea	373
	16.2.32 SetMotionPara	374
	16. 2. 33 SetPassengerLinePara	375
	16. 2. 34 SetPassengerOSDPara	376
	16. 2. 35 SetPassengerPara	376
	16. 2. 36 SetRecordPlanPara	377
	16. 2. 37 SetRS232Para	378
	16. 2. 38 SetRS485Para	379
	16. 2. 39 SetVILostPara	380
16.	.3 XML或JSON报文	381
	16.3.1 JSON_DasInfo	381
	16.3.2 JSON_EventNotificationAlert_FaceCaptureMsg	381
	16.3.3 JSON_EventNotificationAlert_FaceModelingMsg	388
	16.3.4 JSON_EventNotificationAlert_FacePicComparisonMsg	394
	16.3.5 JSON_EventNotificationAlert_HeatMapMsg	398
	16.3.6 JSON_EventNotificationAlert_MultiTargetTypeDetectMsg	399
	16.3.7 JSON_EventNotificationAlert_PeopleQueuingUpMsg	408
	16.3.8 JSON_EventNotificationAlert_UnregisteredStreetVendorMsg	410
	16.3.9 JSON_EventNotificationAlert_WaitingTimeDetectionMsg	413
	16.3.10 XML_EventNotificationAlert_AlarmInputMsg	416
	16.3.11 XML_EventNotificationAlert_ANPRMsg	417
	16.3.12 XML_EventNotificationAlert_AudioExceptionMsg	422
	16.3.13 XML_EventNotificationAlert_DefocusDetectionMsg	423
	16.3.14 XML_EventNotificationAlert_FaceDetectionMsg	424
	16.3.15 XML_EventNotificationAlert_FastMovingDetectionMsg	424
	16.3.16 XML_EventNotificationAlert_HDDErrorAlarmMsg	425
	16.3.17 XML_EventNotificationAlert_IllegalLoginAlarmMsg	426
	16 3 18 XML EventNotificationAlert IntrusionMsg	426

16.3.19 XML_EventNotificationAlert_LineCrossingMsg	427
16.3.20 XML_EventNotificationAlert_LoiteringDetectionMsg	428
16.3.21 XML_EventNotificationAlert_MotionDetectionMsg	429
16.3.22 XML_EventNotificationAlert_ObjectRemovalDetectionMsg	430
16.3.23 XML_EventNotificationAlert_ParkingDetectionMsg	431
16.3.24 XML_EventNotificationAlert_PeopleCountingEventMsg	432
16.3.25 XML_EventNotificationAlert_PeopleGatheringDetectionMsg	432
16.3.26 XML_EventNotificationAlert_PIRAlarmMsg	433
16.3.27 XML_EventNotificationAlert_RegionEntranceMsg	434
16.3.28 XML_EventNotificationAlert_RegionExitingMsg	435
16.3.29 XML_EventNotificationAlert_SceneChangeDetectionMsg	436
16.3.30 XML_EventNotificationAlert_TemperatureAlarmMsg	437
16.3.31 XML_EventNotificationAlert_UnattendedBaggageDetectionMsg	438
16.3.32 XML_EventNotificationAlert_VideoLossMsg	439
16.3.33 XML_EventNotificationAlert_VideoTamperingMsg	440
附录 A. 附录	441
A.1 事件类型和详情	441
A. 2 ISUPSDK 错误码	443
A.3 设备日志类型	447
A.3.1 报警主机日志类型	463
A.4 支持的网络摄像机	472
A 5 支持的网络硬盘录像机	480

## 第1章 概览

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。

### 1.1 简介

ISUPSDK 是基于设备主动向平台或服务注册的智能安全上行协议(Intelligent Security Uplink Protocol,可简称为 ISUP)进行交互和通信的,因此常适用于无固定 IP 地址的设备。该 SDK 提供了多个接口和命令以实现不同的功能,同时便于更多第三方平台的接入,简化了集成过程。

### 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。

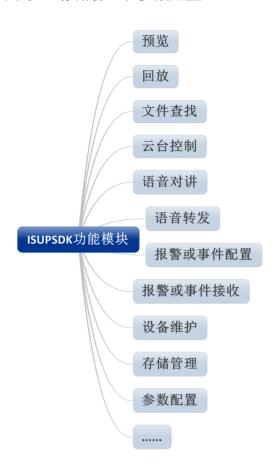


图 1-1 ISUPSDK 功能模块

### 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 Serivice,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。

### Ţi 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. d11
	HCISUPCMS. 1ib
	HCISUPCMS. h
取流模块	HCISUPStream. dl1
	HCISUPStream.lib
	HCISUPStream.h
报警监听模块	HCISUPAlarm. dll
	HCISUPAlarm.lib
	HCISUPAlarm.h
存储管理模块	HCISUPSS. d11
	HCISUPSS. 1ib
	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. 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。

#### 操作步骤

### li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  $\mathbf{n}$  **配**  $\mathbf{m}$   $\mathbf{$
- 3. 选择平台接入方式为 /SUP。
- 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。

### i说明

密钥只有在 ISUP 版本选择 5.0 时是必须配置的。

- 6. 通过 ISUPSDK 调用 NET\_ECMS\_Init 初始化 CMS。
- 7. 通过 ISUPSDK 调用 NET\_ECMS\_StartListen 开启监听并注册回调函数。

#### 「」 道。 说明

回调函数需要按照一定的逻辑流程进行注册,详见 *回调函数的注册流程*。

### 2.1 回调函数的注册流程

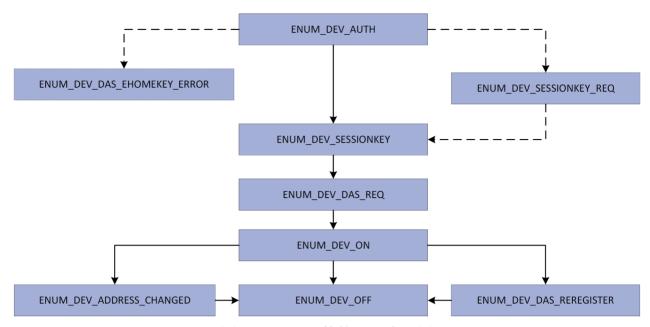


图 2-1 回调函数的注册流程图

#### 示例

回调函数注册流程的示例代码

```
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] = "i#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

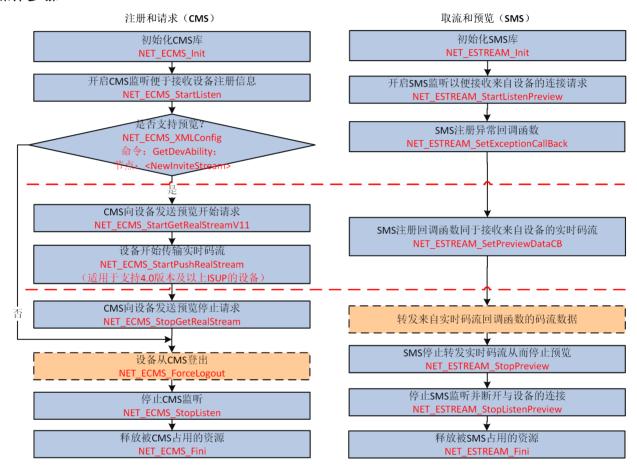


图 3-1 预览的接口调用流程图

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\_SetExceptionCal | Back* 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。

### [\_i]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

### Ţi 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 1LoginID = -1;
LONG 1LinkHandle = -1;
LONG 1RealHandle = -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)
```

```
1LoginID = 1UserID;
         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 (1UserID);
   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 1Port))
          return FALSE;
       if (!PlayM4 SetStreamOpenMode(m 1Port, STREAME REALTIME))
          return FALSE;
       //输入头部的前 40 字节
       if(!PlayM4 OpenStream(m 1Port, (unsigned char *)pBuffer, (DWORD)iDataLen, 2*1024*1024))
          return FALSE;
       if(!PlayM4_Play(m_1Port, hWnd))
          return FALSE;
   else
       for (int i=0; i<1000; i++)
          BOOL bRet = PlayM4 InputData(m 1Port, (unsigned char *) pBuffer, (DWORD) iDataLen);
          if (!bRet)
              if (i >= 999)
                 printf("PlayM4 InputData failed, error code: %d!",
PlayM4 GetLastError(m 1Port));
              S1eep(2);
   return TRUE;
//注册实时码流回调函数
void CALLBACK fnPREVIEW DATA CB (LONG 1PreviewHandle, NET EHOME PREVIEW CB MSG *pPreviewCBMsg,
void *pUserData)
if (NULL == pPreviewCBMsg)
```

```
return ;
   1RealHandle = 1PreviewHandle;
   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 1PreviewHandle, NET EHOME NEWLINK CB MSG
*pNewLinkCBMsg, void *pUserData)
   1LinkHandle = 1PreviewHandle:
   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(1PreviewHandle, &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; //O-TCP, 1-UDP
   //开启监听服务
   LONG 1Handle = NET ESTREAM StartListenPreview(&struListen);
   if (1Handle < -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 1Listen = 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(1LoginID >= 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; //O-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.1SessionID))
      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 (1Handle \geq = 0)
      if (!NET_ESTREAM_StopListenPreview(1Handle))
          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

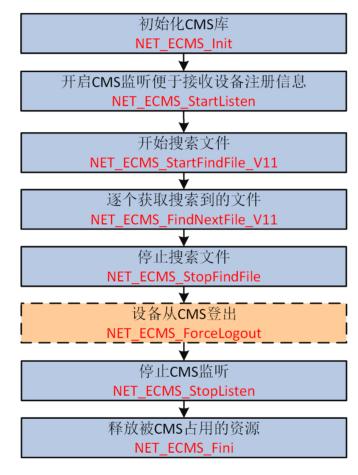


图 4-1 文件查找的接口调用流程图

- 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 1LoginID = -1:
LONG 1LinkHandle = -1:
LONG 1PlayHandle = -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 1Listen = NET ECMS StartListen(&struCMSListenPara);
if (1Listen < -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(1LoginID >= 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 1SearchType = 0; //查找视频文件
LONG 1FileHandle = NET_ECMS_StartFindFile_V11(1LoginID, 1SearchType, &struFindCond,
sizeof(struFindCond)); //开始文件查找
if (1FileHandle < 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 1Ret = -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 (1Ret == 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 (1Ret == ENUM GET NETX STATUS NEED WAIT)
           S1eep(5);
           continue;
       if ((1Ret == ENUM GET NETX STATUS NO FILE) | (1Ret == ENUM GET NEXT STATUS FINISH))
           printf("No more file!\n");
           break;
       else if(1Ret == 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();
```

- 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

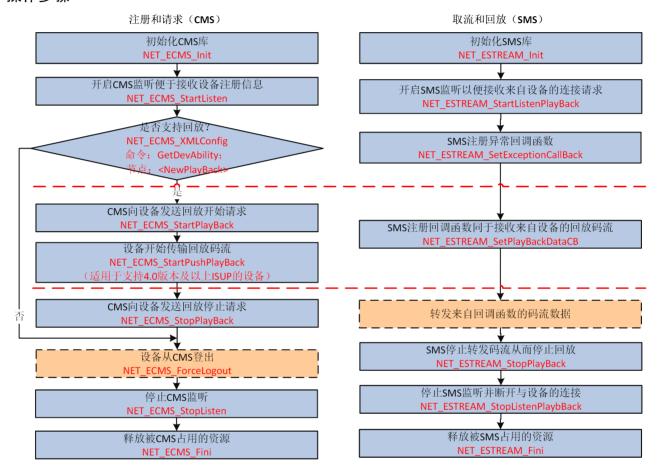


图 5-1 回放的接口调用流程图

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。

## ISUPSDK (通用) 开发指南

- 3. 调用 *NET\_ESTREAM\_SetExceptionCal | Back* 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。

#### [〕 道说明

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

# ii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。

```
if (pDevInfo != NULL)
         1LoginID = 1UserID;
         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(1UserID);
   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_1Port))
          return FALSE;
       if (!PlayM4 SetStreamOpenMode(m 1Port, STREAME REALTIME))
          return FALSE;
       //Enter first 40-byte header
       if(!PlayM4_OpenStream(m_1Port, (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 1Port, (unsigned char *) pBuffer, (DWORD) iDataLen);
          if (!bRet)
             if (i >= 999)
                 printf("PlayM4 InputData failed, error code: %d!",
PlayM4_GetLastError(m_1Port));
             S1eep(2);
   */
   return TRUE;
//注册回放码流回调函数
BOOL CALLBACK fnPLAYBACK DATA CB(LONG 1PlayBackLinkHandle, NET EHOME PLAYBACK DATA CB INFO
*pDataCBInfo, void *pUserData)
   if (NULL == pDataCBInfo)
      return FALSE;
   1PlayHandle = 1PlayBackLinkHandle;
   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 1PlayBackLinkHandle,
NET_EHOME_PLAYBACK_NEWLINK_CB_INFO *pNewLinkCBMsg, void *pUserData)
   1LinkHandle = 1PlayBackLinkHandle;
   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(!PlayBackLinkHandle, &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; //O-TCP, 1-UDP (保留)
```

```
//开启监听服务
   LONG 1Handle = 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 1Listen = NET ECMS StartListen(&struCMSListenPara);
   if (1Listen < -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 1SearchType = 0; //查找文件
    LONG 1FileHandle = NET ECMS StartFindFile V11(1LoginID, 1SearchType, &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 1Ret = -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 (1Ret == 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],
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 (1Ret == ENUM GET NETX STATUS NEED WAIT)
               S1eep(5);
               continue;
           if ((1Ret == ENUM GET NETX STATUS NO FILE) || (1Ret == 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. 1SessionID = struPlayBackOut. 1SessionID; //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(1PlayHandle))
    {
        printf("NET_ESTREAM_StopPlayBack failed, error code: %d\n",
NET_ECMS_GetLastError());
    }
}

//SMS 停止监听服务
    if (!LinkHandle >= 0)
{
        if (!NET_ESTREAM_StopListenPlayBack(1LinkHandle))
        {
            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");
}
```

- 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

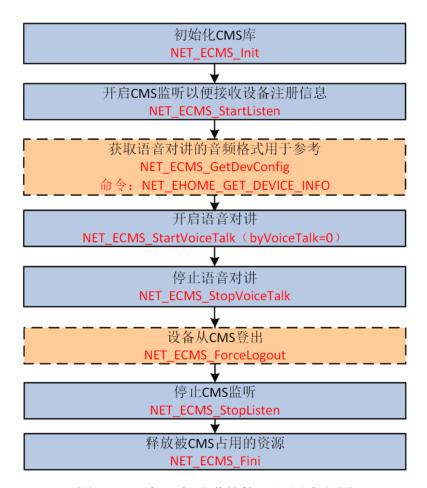


图 6-1 开启语音对讲的接口调用流程图

# 〕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 1LoginID = -1:
LONG m 1VoiceTalkHandle = -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)
         1LoginID = 1UserID;
         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 (1UserID);
```

```
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 1Listen = 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");
```

- 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

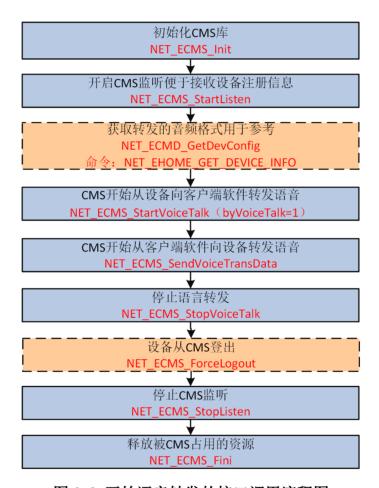


图 6-2 开始语音转发的接口调用流程图

- 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。

- 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

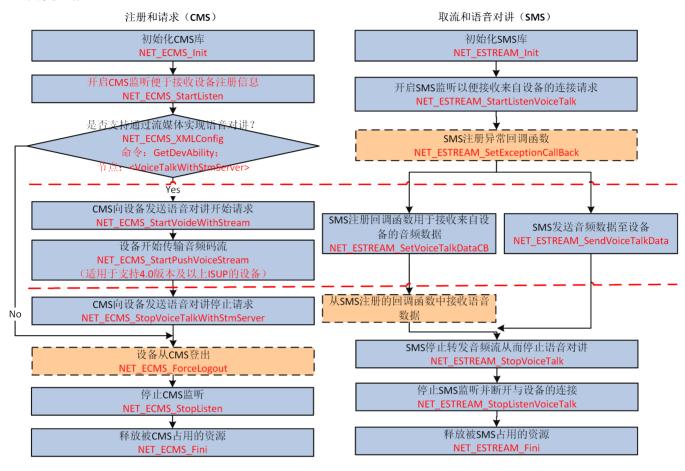


图 6-3 通过流媒体服务开启语音对讲的接口调用流程图

# li 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 SetExceptionCal/Back 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。

#### l 道说明

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

# ´Ţij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。

```
1LoginID = 1UserID;
        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(1UserID);
   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 1Handle, NET EHOME VOICETALK DATA CB INFO *pDataCBInfo,
void *pUserData)
   if (NULL == pDataCBInfo)
      return ;
   1VoiceHandle = 1Handle;
   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 1LinkHandle, NET EHOME VOICETALK NEWLINK CB INFO
*pNewLinkCBInfo, void *pUserData)
   1VoiceLinkHandle = 1LinkHandle;
   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(!VoiceLinkHandle, &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 1pArg)
   LONG 1Hanle = (LONG) 1pArg;
   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 1Handle = NET ESTREAM StartListenVoiceTalk(&struListen);
   if (1Handle < -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 1Listen = 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(1LoginID >= 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
   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. 1SessionID = struVoiceTalkOut. 1SessionID;
   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, O, LPTHREAD START ROUTINE(TestVoiceTransThread),
(void*) 1VoiceLinkHandle, 0, &dwThreadId);
   Sleep (50000);
   //退出
   //释放 CMS 语音对讲请求资源
   if(!NET ECMS StopVoiceTalkWithStmServer(lLoginID, struVoiceTalkOut.1SessionID))
       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");
}
```

- 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 NET_ECMS_RemoteControl,将 dwCommand 设置为"NET_EHOME_PTZ_CTRL"(命令号: 1000),并将 NET_EHOME_REMOTE_CTRL_PARAM 中的 lpInbuffer 设置为 NET_EHOME_PTZ_PARAM。
配置、调用和清除预置点	调用 NET_ECMS_RemoteControl,将 dwCommand 设置为"NET_EHOME_PRESET_CTRL"(命令号: 1001),并将 NET_EHOME_REMOTE_CTRL_PARAM 中的 lpInbuffer 设置为 NET_EHOME_PRESET_PARAM。
执行 3D 定位	调用 NET_ECMS_XMLConfig 传输命令报文 PZIN 。

# 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。

# 

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 *开启监听服务并接收报警*)。有关报警或事件详情,请参见 *表 15-1* 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 NET_ECMS_XMLConfig 传输命令报文 GetVILostPara 。
设置视频丢失侦测参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetVILostPara 。

# 隐私遮蔽

## 表 8-4 隐私遮蔽

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

# 异常报警

# 表 8-5 异常报警

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

# 客流统计

# 表 8-6 客流统计

获取客流统计参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetPassengerPara 。
设置客流统计参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetPassengerPara 。
获取客流统计信息 OSD 参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetPassengerOSDPara 。
配置客流统计信息 OSD 参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetPassengerOSDPara 。
获取客流统计检测线参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetPassengerLinePara 。
配置客流统计检测线参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetPassengerLinePara 。

# ISUPSDK(通用) 开发指南

获取客流统计摄像机标定参	调用 NET_ECMS_XMLConfig 传输命令报文
数	GetCalibrationPara 。
配置客流统计摄像机标定参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetCalibrationPara 。

# 报警输入

# 表 8-7 报警输入

为"NET_
ndBuf 设
为"NET_
Buf 设置
Ī
]

# 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 NET_ECMS_XMLConfig 传输命令报文
	GetAlarmDeploymentTime 。
配置报警布防计划参数	调用 NET_ECMS_XMLConfig 传输命令报文
	SetAlarmDeploymentTime 。

# 报警联动

## 表 8-9 报警联动

获取报警联动参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetAlarmLinkageType 。
配置报警联动参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetAlarmLinkageType 。
获取抓拍联动参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetAlarmTriggerCapture 。

# ISUPSDK(通用) 开发指南

配置抓拍联动参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetAlarmTriggerCapture 。
获取录像联动参数	调用 NET_ECMS_XMLConfig 传输命令报文 GetAlarmTriggerRecord 。
配置录像联动参数	调用 NET_ECMS_XMLConfig 传输命令报文 SetAlarmTriggerRecord 。
手动控制报警输出联动	调用 NET_ECMS_RemoteControl,将 dwCommand 设置为"NET_EHOME_MANUAL_IOOUT"(命令号: 19),并将 NET_EHOME_REMOTE_CTRL_PARAM 中的 lpInbuffer 设置为 NET_EHOME_MANUAL_IOOUT_CTRL。
获取报警输出参数	调用 NET_ECMS_GetDevConfig ,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 NET_EHOME_ALARMOUT_CFG 返回。
配置报警输出参数	调用 NET_ECMS_SetDevConfig ,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 NET_EHOME_ALARMOUT_CFG 。

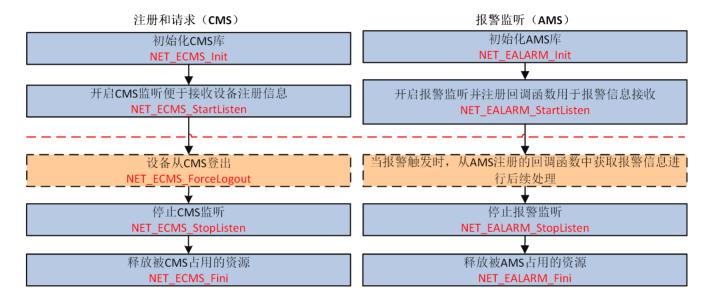
# 第9章 开启监听服务并接收报警

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 开启临听服务并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。

# □ii说明

SessionKey 通过回调数据类型 "enum\_dev\_sessionkey" 获得,详见 *回调函数的注册流程* 。

- 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

开启监听服务并接收报警的示例代码

# 

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 1LoginID = -1;
LONG 1ListenHandle = -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)
        1LoginID = 1UserID;
        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 (1UserID);
   else
   return TRUE;
//AMS 报警回调函数
BOOL CALLBACK AlarmMSGCallBack (LONG 1Handle, NET EHOME ALARM MSG *pAlarmMsg, void *pUserData)
```

```
1ListenHandle = 1Handle;
   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 1Handle = 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 1Listen = 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;
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!");
```

- 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 NET_ECMS_GetDevConfig ,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 NET_EHOME_DEVICE_CFG 返回。	
配置设备信息	调用 NET_ECMS_SetDevConfig ,将 dwCommand 设置为"NET_EHOME_SET_DEVICE_CFG"(命令号: 4),并将 pInBuf 设置为NET_EHOME_DEVICE_CFG。	
查询设备工作状态	调用 NET_ECMS_XMLRemoteControl 传输命令报文 GETDEVICEWORKSTATUS 。	
重启设备	调用 NET_ECMS_XMLRemoteControl 传输命令报文 REBOOT 。	
升级设备	调用 NET_ECMS_XMLRemoteControl 传输命令报文 UPDATE 。	
执行手动校时	调用 NET_ECMS_XMLRemoteControl 传输命令报文 ADJUSTTIME 。	
获取设备版本信息	调用 NET_ECMS_GetDevConfig ,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 NET_EHOME_DEVICE_CFG 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。

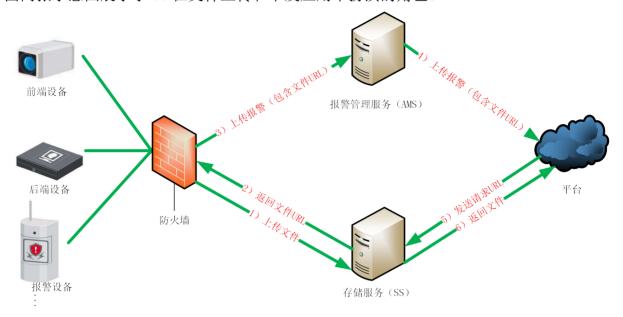


图 11-1 设备上传文件至平台

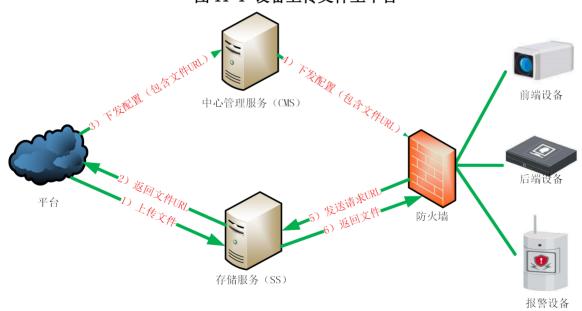


图 11-2 平台下发文件至设备

# 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。 操作步骤

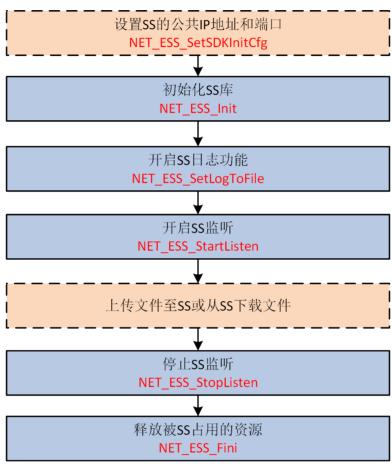


图 11-3 开启存储服务的接口调用流程图

1. 可选操作: 调用 NET\_ESS\_SetSDKInitCfg , 设置 enumType 为 "NET\_EHOME\_SS\_INIT\_CFG\_PUBLIC\_IP\_PORT",并将 lpInBuff 设置为 NET\_EHOME\_IPADDRESS , 从而设置 SS 的公共 IP 地址和端口。

# []i]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。
- 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 struSSListenParam = { 0 };
```

```
memcpy(struSSListenParam.struAddress.szIP, "10.8.97.60", strlen("10.8.97.60"));
    struSSListenParam.struAddress.wPort = 8080;
    memcpy(struSSListenParam.szKMS UserName, "test", strlen("test"));
memcpy(struSSListenParam.szKMS_Password, "12345", strlen("12345"));
memcpy(struSSListenParam.szAccessKey, "test", strlen("test"));
memcpy(struSSListenParam.szSecretKey, "12345", strlen("12345"));
    struSSListenParam.fnSMsgCb = SS Message Callback;
    struSSListenParam.fnSStorageCb = SS Storage Callback;
    struSSListenParam.pUserData = NULL;
    long m 1SSHandle = NET ESS StartListen(&struSSListenParam);
    if (-1 == m 1SSHandle)
        printf("NET ESS StartListen Failed, port: %d", struSSListenParam.struAddress.wPort);
    else
        printf("NET ESS StartListen succ port: %d", struSSListenParam.struAddress.wPort);
    char cTmp = ' \setminus 0';
    do
        printf("Input q to exit!");
        cTmp = getchar();
    } while (cTmp != 'q');
    //停止监听
    NET ESS StopListen (m 1SSHandle);
    //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。

#### 前提条件

请确保已开启 SS,详见 *开启存储服务* 。

### 操作步骤

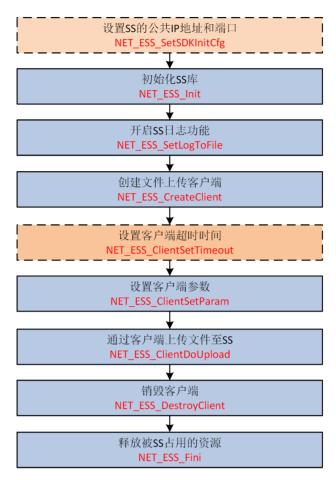


图 11-4 上传文件至存储服务的接口调用流程图

1. 可选操作: 调用 NET\_ESS\_SetSDKInitCfg , 设置 enumType 为 "NET\_EHOME\_SS\_INIT\_CFG\_PUBLIC\_IP\_PORT",并将 lpInBuff 设置为 NET\_EHOME\_IPADDRESS , 从而设置 SS 的公共 IP 地址和端口。

# 「<mark>」</mark>道说明

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。
- 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。
- 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. 完成下载之后,调用 NET\_ESS\_DestroyClient 销毁上传客户端。

### 示例

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
   1SSClientHandle = NET ESS CreateClient(&struClientParam);
    if (1SSClientHandle >= 0)
       NET ESS ClientSetTimeout (ISSClientHandle, 60 * 1000, 60 * 1000);
       NET ESS ClientSetParam(1SSClientHandle, SS CLIENT FILE PATH PARAM NAME, "C:\\Picture\
1. jpg'';
       char szUrl[MAX URL LEN SS] = { 0 };
       NET ESS ClientDoUpload(ISSClientHandle, szUrl, MAX URL LEN SS - 1);
       NET ESS DestroyClient(1SSClientHandle);
    //基于 VRB 协议上传文件至 SS
    struClientParam.enumType = NET EHOME SS CLIENT TYPE VRB;
    1SSClientHandle = NET ESS CreateClient(&struClientParam);
    if (1SSClientHandle >= 0)
       NET ESS ClientSetTimeout (ISSClientHandle, 60 * 1000, 60 * 1000);
       NET ESS ClientSetParam(1SSClientHandle, SS CLIENT FILE PATH PARAM NAME, "C:\\Picture\
1. jpg'';
       NET ESS ClientSetParam(ISSClientHandle, SS CLIENT VRB FILENAME CODE,
"filename=1.jpg&code=");
       NET ESS ClientDoUpload(ISSClientHandle, szUrl, MAX URL LEN SS - 1);
       NET ESS DestroyClient(1SSClientHandle);
   //基于 KMS 协议上传文件至 SS
```

```
struClientParam.enumType = NET EHOME SS CLIENT TYPE KMS;
   1SSClientHandle = NET ESS CreateClient(&struClientParam);
    if (1SSClientHandle >= 0)
       NET ESS ClientSetTimeout (ISSClientHandle, 60 * 1000, 60 * 1000);
       NET ESS ClientSetParam(1SSClientHandle, SS CLIENT FILE PATH PARAM NAME, "C:\\Picture\
1. jpg'';
       NET ESS ClientSetParam(ISSClientHandle, SS CLIENT KMS USER NAME, "test");
       NET ESS ClientSetParam(ISSClientHandle, SS CLIENT KMS PASSWIRD, "12345");
       NET ESS ClientDoUpload(ISSClientHandle, szUrl, MAX URL LEN SS - 1);
       DWORD dwErr = NET ESS GetLastError();
       NET ESS DestroyClient(1SSClientHandle);
   //基于 ISUP5.0 上传文件至 SS
    struClientParam.enumType = NET EHOME SS CLIENT TYPE CLOUD;
    1SSClientHandle = NET ESS CreateClient(&struClientParam);
    if (1SSClientHandle >= 0)
       NET ESS ClientSetTimeout (ISSClientHandle, 60 * 1000, 60 * 1000);
       NET ESS ClientSetParam(1SSClientHandle, SS CLIENT FILE PATH PARAM NAME, "C:\\Picture\
\1. jpg");
        NET ESS ClientSetParam(1SSClientHandle, SS CLIENT CLOUD AK NAME, "test");
       NET ESS ClientSetParam(1SSClientHandle, SS CLIENT CLOUD SK NAME, "12345");
       NET ESS ClientDoUpload(ISSClientHandle, szUrl, MAX URL LEN SS - 1);
       DWORD dwErr = NET ESS GetLastError();
       byte* bFileContent = NULL;
       NET ESS DestroyClient(1SSClientHandle);
    //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。

#### 前提条件

请确保已开启 SS, 详见 *开启存储服务*。

### 操作步骤

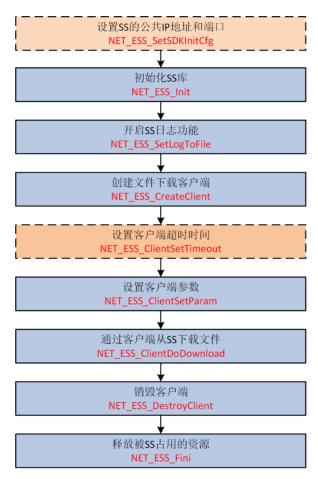


图 11-5 从存储服务下载文件的接口调用流程图

1. 可选操作: 调用 NET\_ESS\_SetSDKInitCfg , 设置 enumType 为 "NET\_EHOME\_SS\_INIT\_CFG\_PUBLIC\_IP\_PORT",并将 1pInBuff 设置为 NET\_EHOME\_IPADDRESS , 从而设置 SS 的公共 IP 地址和端口。

# 道说明

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。
- 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。
- 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

从存储服务器下载文件的示例代码

```
#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 协议和 VRB 协议从 SS 下载文件
   struClientParam.enumType = NET EHOME SS CLIENT TYPE VRB;
    1SSClientHandle = NET ESS CreateClient(&struClientParam);
    if (1SSClientHandle >= 0)
       NET ESS ClientSetTimeout (ISSClientHandle, 60 * 1000, 60 * 1000);
       byte* bFileContent = NULL;
       DWORD dwFileLen = 0;
       if (NET ESS ClientDoDownload(1SSClientHandle, szUrl, (void**)&bFileContent, dwFileLen))
           FILE* pFile = fopen("C://Picture//vrb.jpg", "wb+");
           if (pFile != NULL)
                fwrite(bFileContent, 1, dwFileLen, pFile);
               fclose(pFile);
       //销毁下载客户端
       NET ESS DestroyClient(1SSClientHandle);
    //基于 KMS 协议从 SS 下载文件
   1SSClientHandle = NET ESS CreateClient(&struClientParam);
```

```
if (1SSClientHandle >= 0)
    NET ESS ClientSetTimeout (ISSClientHandle, 60 * 1000, 60 * 1000);
    NET ESS ClientSetParam(ISSClientHandle, SS CLIENT KMS USER NAME, "test");
    NET_ESS_ClientSetParam(ISSClientHandle, SS_CLIENT KMS PASSWIRD, "12345");
    byte* bFileContent = NULL;
    DWORD dwFileLen = 0;
    if (NET ESS ClientDoDownload(1SSClientHandle, 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(1SSClientHandle);
//基于 5.0 版本 ISUP 从 SS 下载文件
struClientParam.enumType = NET EHOME SS CLIENT TYPE CLOUD;
1SSClientHandle = NET ESS CreateClient(&struClientParam);
if (1SSClientHandle >= 0)
    NET ESS ClientSetTimeout (ISSClientHandle, 60 * 1000, 60 * 1000);
    NET ESS ClientSetParam(ISSClientHandle, SS CLIENT CLOUD AK NAME, "test");
    NET ESS ClientSetParam(1SSClientHandle, SS CLIENT CLOUD SK NAME, "12345");
    byte* bFileContent = NULL;
    DWORD dwFileLen = 0;
    if (NET ESS ClientDoDownload(1SSClientHandle, 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(1SSClientHandle);
//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。

# 网络配置

### 表 12-1 网络配置

获取网络参数	调用 NET_ECMS_GetDevConfig ,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 NET_EHOME_NETWORK_CFG 返回。
配置网络参数	调用 NET_ECMS_SetDevConfig ,将 dwCommand 设置为"NET_EHOME_SET_NETWORK_CFG"(命令号: 6),并将 pInBuf 设置为NET_EHOME_NETWORK_CFG。

# 编码配置

### 表 12-2 编码配置

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

# 图像和 OSD 配置

# 表 12-3 图像和 OSD 配置

获取 OSD 参数	调用 NET_ECMS_GetDevConfig ,将 dwCommand 设置为"NET_
	EHOME_GET_PIC_CFG"(命令号: 23)。

	OSD 参数由 pOutBuf 通过结构体 NET_EHOME_PIC_CFG 返回。
配置 OSD 参数	调用 NET_ECMS_SetDevConfig ,将 dwCommand 设置为"NET_EHOME_SET_PIC_CFG"(命令号: 24),并将 pInBuf 设置为NET_EHOME_PIC_CFG。
获取图像参数	调用 NET_ECMS_GetDevConfig ,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 NET_EHOME_IMAGE_CFG 返回。
配置图像参数	调用 NET_ECMS_SetDevConfig ,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 NET_EHOME_IMAGE_CFG 。

# 录像配置

# 表 12-4 录像配置

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

# 串口配置

# 表 12-5 串口配置

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

# 其他配置

# 表 12-6 其他配置

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

# 第13章 接口参考

# 13.1 注册模块

# 13.1.1 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。

# 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 1UserID
);
```

# 参数

#### 1UserID

[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

### 参数

### enumType

[IN] 初始化配置类型,详见枚举 NET\_EHOME\_CMS\_INIT\_CFG\_TYPE 。

### 1pInBuff

[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。
- 多个模块用于同一线程时,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)用于同一线程,如果 CMS 己加载 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 将使用己加载的 OpenSSL。

## 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:0x00030000(3.0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 (默认), FALSE。

## 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\\"。
- 可反复调用该接口以创建多个新的日志文件。最多可创建 10 个日志文件。
- 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。

# NET\_ECMS\_GetSDKLocalCfg

获取中心管理服务器(CMS)的本地配置参数。

# 接口定义

### 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* 。

### 1pOutBuff

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

enumType	1pOutBuff	
ACTIVE_ACCESS_SECURITY	NET_EHOME_LOCAL_ACCESS_SECURITY	
AMS_ADDRESS	NET_EHOME_AMS_ADDRESS	
SEND_PARAM	待发送的数据	
SET_REREGISTER_MODE	重复注册模式	
LOCAL_CFG_TYPE_GENERAL	NET_EHOME_LOCAL_GENERAL_CFG	
COM_PATH	COM 路径。	
SESSIONKEY_REQ_MOD	"True"(回调) 或 "False"(不回调)	

## 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

## 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* 。

### 1pInBuff

[IN] 本地配置参数,与本地配置类型(enumType)相关,详见下表。

enumType	1pOutBuff	
ACTIVE_ACCESS_SECURITY	NET_EHOME_LOCAL_ACCESS_SECURITY	
AMS_ADDRESS	NET_EHOME_AMS_ADDRESS	
SEND_PARAM	待发送的数据	
SET_REREGISTER_MODE	重复注册模式	
COM_PATH	COM 路径。	
SESSIONKEY_REQ_MOD	"True"(回调) 或 "False"(不回调)	

## 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。

# 13.1.2 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。

### 参见

NET\_ECMS\_GetDeviceSessionKey

# NET\_ECMS\_GetDevConfig

获取设备参数。

# 接口定义

```
BOOL CALLBACK NET_ECMS_GetDevConfig(
LONG 1UserID,
DWORD dwCommand,
NET_EHOME_CONFIG *1pConfig,
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。

### 1pConfig

{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] 1pConfig 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	IpConfigp → pCondBuf	<i>lpConfigp</i> → <i>pOutBuf</i>	命令号
获取设备信息的命令: NET_ EHOME_GET_DEVICE_INFO	NULL	NET_EHOME_DEVICE_INF O	1
获取版本信息的命令: NET_ EHOME_GET_VERSION_INFO	NULL	NET_EHOME_VERSION_IN FO	2

dwCommand	lpConfigp → pCondBuf	/pConfigp → pOutBuf	命令号
获取设备基本信息的命 令: NET_EHOME_GET_ DEVICE_CFG	NULL	NET_EHOME_DEVICE_CFG	3
获取网络参数的命令: NET_ EHOME_GET_NETWORK_CFG	NULL	NET_EHOME_NETWORK_CF G	5
获取编码参数的命令: NET_ EHOME_GET_COMPRESSION_ CFG	NET_EHOME_COMPRESSI ON_COND	NET_EHOME_COMPRESSIO N_CFG	7
获取图像参数的命令: NET_ EHOME_GET_IMAGE_CFG	4字节通道号	NET_EHOME_IMAGE_CFG	9
获取报警输入参数的命 令: NET_EHOME_GET_ ALARMIN_CFG	NET_EHOME_ALARMIN_C OND	NET_EHOME_ALARMIN_CF G	11
获取布防计划参数的命 令: NET_EHOME_GET_ALARM_ TIME_CFG	NET_EHOME_ALARM_TIM E_COND	NET_EHOME_ALARM_TIME _CFG	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。	NET_EHOME_ALARMOUT_C FG	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。	NET_EHOME_ALARMOUT_S TATUS_CFG	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	NET_EHOME_PIC_CFG	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 1UserID,
DWORD dwCommand,
NET_EHOME_CONFIG *1pConfig,
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。

### 1pConfig

{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] 1pConfig 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	<i>lpConfigp → pCondBuf</i>	<i>lpConfigp</i> → p <i>lnBuf</i>	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_CF G	6

dwCommand	IpConfigp → pCondBuf	<i>lpConfigp</i> → <i>pInBuf</i>	命令号
设置编码参数的命令: NET_ EHOME_SET_COMPRESSION_ CFG	NET_EHOME_COMPRESSI ON_COND	NET_EHOME_COMPRESSIO N_CFG	8
设置图像参数的命令: NET_ EHOME_SET_IMAGE_CFG	4 字节通道号	NET_EHOME_IMAGE_CFG	10
设置报警输入参数的命 令: NET_EHOME_SET_ ALARMIN_CFG	NET_EHOME_ALARMIN_C OND	NET_EHOME_ALARMIN_CF G	12
设置布防计划参数的命 令: NET_EHOME_SET_ALARM_ TIME_CFG	NET_EHOME_ALARM_TIM E_COND	NET_EHOME_ALARM_TIME _CFG	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。	NET_EHOME_ALARMOUT_C FG	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。	NET_EHOME_ALARMOUT_S TATUS_CFG	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	NET_EHOME_PIC_CFG	23

# 参见

NET\_ECMS\_GetDevConfig

# ${\bf NET\_ECMS\_ResetDevRegisterState}$

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 1UserID,
DWORD dwCommand,
NET_EHOME_REMOTE_CTRL_PARAM *1pCtrlParam
);
```

## 参数

### 1UserID

[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
NET_EHOME_PZIN	放大或缩小。	1002
NET_EHOME_PTRACK	3D 定位。	1003

### 1pCtrlParam

[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

### 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

# 参数

#### 1User1D

「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 1Handle
)
```

# 参数

1Handle

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 1Handle,
NET_EHOME_LONG_CFG_SEND *pSend
)
```

### 参数

#### 1Handle

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\_MakelFrame

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。

# NET\_ECMS\_STDBinaryCtrl

控制二进制参数。

# 接口定义

```
BOOL CALLBACK NET_ECMS_STDBinaryCtr1(
LONG 1UserID,
LONG dwCommand,
NET_EHOME_BINARY_SEND_DATA *pSendData
)
```

## 参数

### 1UserID

[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	OxD	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

# 参数

### 1UserID

[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。

命令	命令号	描述
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。
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。
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。

# 13.1.3 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

### 1pCMSListenPara

[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

#### 1Handle

[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

## 1pStru

[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

#### 1ListenHandle

[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

## 13.1.4 预览接口

## NET\_ECMS\_StartGetRealStreamV11

请求开始预览。

## 接口定义

## 参数

#### 1UserID

[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\_V11。

#### 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。

### 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 1UserID,
NET_EHOME_PUSHSTREAM_IN *pPushInfoIn,
NET_EHOME_PUSHSTREAM_OUT *pPushInfoOut
);
```

### 参数

#### 1UserID

[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 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_StopGetRea1StreamEx(
LONG 1UserID,
NET_EHOME_STOPSTREAM_PARAM *pStopParam
)
```

## 参数

#### 1UserID

[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

## 13.1.5 回放接口

## NET\_ECMS\_StartFindFile\_V11

开始查找文件。

### 接口定义

### 参数

#### 1UserID

[IN] 用户 ID, 由 NET\_ECMS\_StartListen 注册的注册回调函数返回。

#### 1SearchType

[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(1SearchType)和检索条件(pFindCond)之间的关系如下表所示。

1SearchType	pFindCond
ENUM_SEARCH_RECORD_FILE	NET_EHOME_REC_FILE_COND
ENUM_SEARCH_PICTURE_FILE	NET_EHOME_PIC_FILE_COND
ENUM_SEARCH_FLOW_INFO	NET_EHOME_FLOW_COND

# ISUPSDK (通用) 开发指南

1SearchType	pFindCond
ENUM_SEARCH_DEV_LOG	NET_EHOME_DEV_LOG_COND
ENUM_SEARCH_ALARM_HOST_LOG	NET_EHOME_ALARM_HOST_LOG_COND

# NET\_ECMS\_FindNextFile\_V11

逐个获取检索到的文件。

## 接口定义

## 参数

#### 1Handle

[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	NET_EHOME_REC_FILE
检索图片文件: ENUM_SEARCH_PICTURE_FILE	NET_EHOME_PIC_FILE
检索流信息: ENUM_SEARCH_FLOW_INFO	NET_EHOME_FLOW_INFO
检索设备日志: ENUM_SEARCH_DEV_LOG	NET_EHOME_DEV_LOG
检索报警主机日志: ENUM_SEARCH_ALARM_HOST_LOG	NET_EHOME_ALARM_HOST_LOG

### 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  $\textit{NET\_ECMS\_StartFindFile\_V11}$  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

### 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

```
NET_EHOME_PUSHPLAYBACK_OUT *pPushInfoOut
);
```

### 参数

#### 1UserID

[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

#### 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

#### 1UserID

[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

# 13.1.6 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 1UserID,
NET_EHOME_VOICE_TALK_IN *1pVoiceTalkIn
NET_EHOME_VOICE_TALK_OUT *1pVoiceTalkOut
);
```

### 参数

#### 1UserID

[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 。

#### 1pVoiceTalkOut

[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 1UserID,
NET_EHOME_PUSHVOICE_IN *1pPushParamIn,
NET_EHOME_PUSHVOICE_OUT *1pPushParamOut
);
```

### 参数

#### 1UserID

[IN] 用户 ID,由 NET\_ECMS\_StartListen 注册的回调函数返回。

#### 1pPushParamIn

[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 。

#### 1pPushParamOut

[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

#### 1UserID

[IN] 用户 ID, 由 NET\_ECMS\_StartListen 注册的回调函数返回。

#### 1SessionID

[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

# 参数

#### 1UserID

「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,

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。

# 13.1.7 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 1UserID,
NET_EHOME_XML_CFG *pXmlCfg,
DWORD dwConfigSize
);
```

### 参数

#### 1UserID

[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 1UserID,
NET_EHOME_XML_CFG *pXmlCfg,
DWORD *pHandle
)
```

#### 参数

#### 1UserID

[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

### 参数

#### 1UserID

[IN] 用户 ID, 由 NET\_ECMS\_StartListen 注册的注册回调函数( DEVICE\_REGISTER\_CB ) 返回。

#### 1pCtrlParam

[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

### 参数

#### iUserID

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

### 1pPTXMLParam

[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 *1pPTXMLParam
);
```

## 参数

#### iUserID

[IN] 用户 ID,由 NET\_ECMS\_StartListen 注册的回调函数返回。

#### 1pPTXMLParam

[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 *1pPTXMLParam
);
```

### 参数

#### iUserID

[IN] 用户 ID, 由 NET\_ECMS\_StartListen 注册的回调函数返回。

#### 1pPTXMLParam

[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 *1pPTXMLParam
);
```

#### 参数

#### iUserID

[IN] 用户 ID,由 NET\_ECMS\_StartListen 注册的回调函数返回。

#### 1pPTXMLParam

[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 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 1UserID,
NET_EHOME_PTXML_PARAM *1pParam
);
```

## 参数

#### 1UserID

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

#### 1pParam

[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

## 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。

## ${\tt NET\_ECMS\_ConvertProtocolHttpToPassthrough}$

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 *1pParam,

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。

#### 1pParam

[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。

# 13.2 报警监听模块

## 13.2.1 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

## 参数

### enumType

[IN] 初始化配置类型,详见枚举 NET\_EHOME\_EALARM\_INIT\_CFG\_TYPE 。

### 1pInBuff

[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。
- 多个模块用于同一线程时,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)用于同一线程,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。

## 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:0x00030000(3.0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

#### nLogLeve1

[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 (默认), FALSE。

### 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\\"。
- 可反复调用该接口以创建多个新的日志文件。最多可创建 10 个日志文件。
- 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。

### 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

## 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

### 1pOutBuff

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

enumType	1pOutBuff
ACTIVE_ACCESS_SECURITY	NET_EHOME_LOCAL_ACCESS_SECURITY
LOCAL_CFG_TYPE_GENERAL	NET_EHOME_LOCAL_GENERAL_CFG
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\_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

### 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* 。

#### 1pInBuff

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

enumType	1pOutBuff
ACTIVE_ACCESS_SECURITY	NET_EHOME_LOCAL_ACCESS_SECURITY
AMS_ADDRESS	NET_EHOME_AMS_ADDRESS
LOCAL_CFG_TYPE_GENERAL	NET_EHOME_LOCAL_GENERAL_CFG

### 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

## 13.2.2 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。

## 参见

NET\_EALARM\_GetDeviceSessionKey

# 13.2.3 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

#### 1ListenHandle

[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。

# 13.3 取流模块

### 13.3.1 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

## 参数

#### enumType

[IN] 初始化配置类型,详见枚举 NET\_EHOME\_ESTREAM\_INIT\_CFG\_TYPE 。

### 1pInBuff

[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。
- 多个模块位于用一线程时,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)用于同一线程,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。

### 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:0x00030000(3.0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

## 参数

#### nLogLeve1

[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"。

#### bAutoDe1

[IN] 是否在一段时间后自动删除日志文件: TRUE (默认), FALSE。

### 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\\"。
- 可反复调用该接口以创建多个新的日志文件。最多可创建 10 个日志文件。
- 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。

## 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

### 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	NET_EHOME_LOCAL_ACCESS_SECURITY
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

## 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 。

#### 1pInBuff

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

enumType	1pOutBuff
ACTIVE_ACCESS_SECURITY	NET_EHOME_LOCAL_ACCESS_SECURITY

## 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

### 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。

### 13.3.2 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

#### 1ListenHandle

[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 1PreviewHandle
);
```

## 参数

#### 1PreviewHandle

[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

## 参数

#### 1Handle

[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。

## ${\tt NET\_ESTREAM\_SetStandardPreviewDataCB}$

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。

## 13.3.3 回放接口

## NET ESTREAM SetPlayBackDataCB

注册回调函数以接收回放码流。

## 接口定义

### 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。

## 13. 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 1Handle,
NET_EHOME_VOICETALK_DATA_CB_PARAM *pStruCBParam
);
```

### 参数

#### 1Handle

[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

#### 1Handle

[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

#### 1ListenHandle

[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 1Handle
);
```

## 参数

#### 1Handle

[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。

## 13.4 存储管理模块

## 13.4.1 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 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\_SetSDKInitCfg

设置存储服务器(SS)初始化参数。

## 接口定义

### 参数

#### enumType

[IN] 初始化配置类型,详见枚举 NET\_EHOME\_SS\_INIT\_CFG\_TYPE 。

### 1pInBuff

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

enumType	lpInBuff
NET_EHOME_SS_INIT_CFG_SDK_PATH	NET_EHOME_SS_LOCAL_SDK_PATH
NET_EHOME_SS_INIT_CFG_CLOUD_TIME_DIFF	时差
NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT	NET_EHOME_IPADDRESS
NET_EHOME_SS_INIT_CFG_LIBEAY_PATH	带有 OpenSSL 加密库库名的路径
NET_EHOME_SS_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。
- 多个模块用于同一线程时,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)用于同一线程,且 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。

## 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"。

#### bAutoDe1

[IN] 是否在一段时间后自动删除日志文件: "TRUE"-是(默认), "FALSE"-否。

#### 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\\"。
- 可反复调用该接口以创建多个新的日志文件。最多可创建 10 个日志文件。
- 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。

## 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

返回版本信息,例如: 0x02000001。

## 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。

## 13.4.2 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。

## 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。

## 13.4.3 上传和下载客户端接口

## 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 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 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

#### 1Handle

[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_PASSWIRD	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 1Handle,
DWORD dwSendTimeout,
DWORD dwRecvTimeout
);
```

## 参数

#### 1Handle

[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

#### 1Handle

[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\_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

#### 1Handle

[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

## 参数

#### 1Handle

[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14章 回调函数

## 14. 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

#### 1pData

[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

## 14. 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,
           *pOutBuffer,
  void
  DWORD
           dw0utLen.
  void
           *pInBuffer,
  DWORD
           dwInLen,
           *pUser
  void
);
```

#### 参数

### 1UserID

[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 **回调函数的注册流程** 。

#### 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。

#### dw0utLen

[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	NET_EHOME_SERVER_INFO	NET_EHOME_DEV_REG_INFO
ENUM_DEV_OFF	无	无
ENUM_DEV_ADDRESS_CHANGED	NET_EHOME_SERVER_INFO	NET_EHOME_DEV_REG_INFO
ENUM_EHOME50_DEV_AUTH	EHomeKey(调用 NET_ECMS_GetDeviceSession Key 获取)	NET_EHOME_DEV_REG_INFO
ENUM_EHOME50_DEV_ SESSIONKEY	无	
ENUM_EHOME50_DEV_DAS_REQ	JSON_DasInfo	

• 回调设备离线信息时,必须调用 NET\_ECMS\_ForceLogout 注销登录。

## 14.3 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。

## 14.4 EHomeMsgCallBack

报警回调函数。

## 回调函数定义

#### 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

## 14.5 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)。

#### dw0utLen

[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

## 14.6 EHomeSSRWCallBack

读写回调函数

### 回调函数定义

## 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

## 14.7 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

## 14.8 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。

#### 1UserID

[OUT] 用户 ID。

#### iHandle

[OUT] 句柄(会话 ID)。

#### pUser

[OUT] 用户参数指针。

## 14.9 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-OggVorbis (Ogg Vorbis), 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。

## 14.10 LongConfigCallBack

长连接回调函数

## 接口定义

### 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。

#### dw0utLen

[OUT] 输出缓冲区中的数据大小。

#### pUser

[OUT] 用户指针。

## 14. 11 PASSTHROUGHDATACALLBACK

数据传输回调函数

## 接口定义

```
typedef void (CALLBACK* PASSTHROUGHDATACALLBACK) (
  DWORD
                  dwProxyType,
  LONG
                  iListenHandle,
                  pDeviceID,
  void
  DWORD
                  dwDevIDLen,
  void
                  pDataBuffer.
                  dwDataLen,
  DWORD
                  pUser
  void
);
```

### 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。

## 14. 12 PLAYBACK\_DATA\_CB

回放数据的回调函数

## 接口定义

## 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。

## 14. 13 PLAYBACK\_NEWLINK\_CB

回放请求的回调函数

### 回调函数定义

### 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

## 14. 14 PREVIEW DATA CB

预览数据的回调函数

## 回调函数定义

### 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。

## 14.15 PREVIEW\_NEWLINK\_CB

接收预览请求响应的回调函数

## 回调函数定义

### 参数

## 1LinkHandle

[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。

## 14. 16 VOICETALK\_DATA\_CB

语音对讲数据的回调函数

## 接口定义

### 参数

#### 1Handle

[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。

## 14. 17 VOICETALK\_NEWLINK\_CB

语音对讲请求的回调函数

## 接口定义

## 参数

#### 1Handle

[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 15 章 结构体和枚举

## 15.1 数据结构体

## 15. 1. 1 NET\_DVR\_TYPE\_NAME

关于硬盘录像机类型和名称的结构体

### 结构体定义

```
struct{
    DWORD dwDVRType;
    char byDevName[24];
}NET_DVR_TYPE_NAME, *LPNET_DVR_TYPE_NAME;
```

## 成员

#### dwDVRType

硬盘录像机类型。

#### byDevName

硬盘录像机名称。

## 15. 1. 2 NET\_EHOME\_ADB\_DEBUG

关于终端 ADB 控制的结构体

## 结构体定义

```
struct {
BOOL bAdbEnable;
BYTE bySecreKey[128];
}NET_EHOME_ADB_DEBUG, *LPNET_EHOME_ADB_DEBUG;
```

#### 成员

#### bAdbEnable

是否开启 ADB 控制。

#### bySecreKey

密钥。

## 15. 1. 3 NET\_EHOME\_ALARM\_HOST\_LOG

关于报警主机日志查询结果的结构体

## 结构体定义

```
struct{
 NET EHOME TIME
                    struLogTime;
  DWORD
                    dwMajorType;
  DWORD
                    dwMinorType:
  DWORD
                    dwParamType:
                    sUserName[NAME LEN/*32*/];
  char
                    sIPAddress[128];
  char
  char
                    sLogContext[MAX LOG INFO LEN/*8 × 1024*/];
  BYTE
                    byTimeDiffH;
  BYTE
                    byTimeDiffM;
                    byRes[62];
  BYTE
NET EHOME ALARM HOST LOG, *LPNET EHOME ALARM HOST LOG;
```

## 成员

#### struLogTime

日志时间信息,详见 NET\_EHOME\_TIME 。

#### dwMajorType

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 报警主机日志类型。

#### dwMinorType

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

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

用户名。

#### sIPAddress

远程主机的 IP 地址。

#### sLogContext

日志内容。

#### bvTimeDiffH

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

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

保留,设为0。

## 15. 1. 4 NET\_EHOME\_ALARM\_HOST\_LOG\_COND

关于报警主机日志查询条件的结构体

## 结构体定义

```
struct{
 DWORD
                     dwMajorType;
  DWORD
                    dwMinorType;
  NET EHOME TIME
                    struStartTime:
  NET EHOME TIME
                    struStopTime;
  DWORD
                    dwStartIndex:
  DWORD
                    dwMaxFileCountPer:
 BYTE
                    byLocalOrUTC:
 BYTE
                    byRes[63];
} NET EHOME ALARM HOST LOG COND, *LPNET EHOME ALARM HOST LOG COND;
```

## 成员

#### dwMajorType

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

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

开始时间,可参见 NET EHOME TIME 。

#### struStopTime

结束时间,可参见 NET\_EHOME\_TIME。

#### dwStartIndex

查询起始位置,从0开始。

#### dwMaxFileCountPer

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

时间类型: 0-设备本地时间, 即设备 OSD 时间; 1-UTC 时间。

#### **byRes**

保留,设为0。

## 15. 1. 5 NET\_EHOME\_ALARM\_INFO

报警信息结构体

## 结构体定义

```
struct{
 DWORD
                                   dwSize:
                                   szAlarmTime[MAX TIME LEN/*32*/];
  char
  char
                                   szDeviceID[MAX DEVICE ID LEN/*256*/];
  DWORD
                                   dwAlarmType;
                                   dwAlarmAction;
  DWORD
  DWORD
                                   dwVideoChannel;
  DWORD
                                   dwAlarmInChannel;
  DWORD
                                   dwDiskNumber;
                                   byRemark[MAX REMARK LEN/*64*/];
  BYTE
  BYTE
                                   bvRetransFlag:
                                   byTimeDiffH;
  BYTE
  BYTE
                                   byTimeDiffM;
  BYTE
                                   byRes1;
                                   szAlarmUploadTime[MAX TIME LEN[/*32*/];
  char
 NET_EHOME_ALARM_STATUS_UNION
                                   uStatusUnion;
                                   byRes2[16];
 BYTE
} NET_EHOME_ALARM_INFO, *LPNET_EHOME_ALARM_INFO;
```

#### 成员

#### dwSize

结构体大小。

#### szAlarmTime

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)。

#### szDeviceID

设备 ID。

#### dwAlarmType

报警类型,参见枚举 EN\_ALARM\_TYPE 。

#### dwAlarmAction

报警状态: 0-开始, 1-停止。

#### dwVideoChannel

视频通道号。

#### dwAlarmInChannel

报警输入号。

#### dwDiskNumber

硬盘号。

#### **byRemark**

备注。

#### byRetransFlag

重传标记: 0-实时包, 1-重传包。

#### byTimeDiffH

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。

#### bvTimeDiffM

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

保留。

#### szAlarmUploadTime

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)。

#### uStatusUnion

状态信息,参见联合体 NET\_EHOME\_ALARM\_STATUS\_UNION。

#### byRes2

保留。

## 15. 1. 6 NET\_EHOME\_ALARM\_ISAPI\_INFO

关于通过 HTTP 上传报警信息的结构体。

#### 结构体定义

```
struct{
  char
           *pAlarmData;
  DWORD
           dwAlarmDataLen;
  BYTE
           byDataType;
  BYTE
           byPicturesNumber;
 BYTE
           byRes[2];
 void
           *pPicPackData;
  BYTE
           byRes1[32];
}NET_EHOME_ALARM_ISAPI_INFO, *LPNET_EHOME_ALARM_ISAPI_INFO;
```

## 成员

#### pAlarmData

报警数据。

#### dwAlarmDataLen

报警数据大小。

#### byDataType

数据类型: 0-无效, 1-XML 格式, 2-JSON 格式。

### byPicturesNumber

图片数量。

## byRes

保留。

### pPicPackData

报警图片数据,参见结构体 NET\_EHOME\_ALARM\_ISAPI\_PICDATA。

## byRes

保留。

## 15. 1. 7 NET\_EHOME\_ALARM\_ISAPI\_PICDATA

关于通过 HTTP 上传的报警图片数据结构体。

## 结构体定义

```
struct {
  DWORD     dwPicLen;
  BYTE     byRes[4];
  char     szFilename;
  BYTE     *pPicData;
} NET_EHOME_ALARM_ISAPI_PICDATA, *LPNET_EHOME_ALARM_ISAPI_PICDATA;
```

## 成员

#### dwPicLen

图片大小。

#### byRes

保留。

#### szFilename

图片文件(包括文件名)的保存路径。

#### pPicData

上传的报警图片数据指针。

## 15. 1. 8 NET\_EHOME\_ALARM\_LISTEN\_PARAM

报警监听参数结构体。

## 结构体定义

```
struct{
  NET_EHOME_IPADDRESS struAddress;
  EHomeMsgCallBack
                       fnMsgCb;
  void
                       *pUserData:
  BYTE
                       byProtocolType;
                       byUseCmsPort;
  BYTE
  BYTE
                       bvUseThreadPool:
  BYTE
                       bvRes1:
                       dwKeepAliveSec;
  DWORD
  DWORD
                       dwTimeOutCount;
                       byRes[20];
  BYTE
}NET_EHOME_ALARM_LISTEN_PARAM, *LPNET_EHOME_ALARM_LISTEN_PARAM;
```

#### 成员

#### struAddress

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。地址格式见结构体 NET EHOME IPADDRESS 。

#### fnMsgCb

[IN]报警回调函数,参见 EHomeMsgCallBack。

#### pUserData

[IN]用户参数。

#### byProtocolType

[IN]协议类型: 0-TCP, 1-UDP, 2-MQTT。

#### byUseCmsPort

[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 NET\_ECMS\_GetSDKLocalCfg 和 NET\_ECMS\_SetSDKLocalCfg 来获取和配置。

#### byUseThreadPool

[IN]回调报警时,是否使用线程池: 0-使用(默认),1-不使用。

#### byRes1

[IN]保留。

### dwKeepAliveSec

[IN]心跳间隔,单位: s。如果设为0,表示默认心跳间隔为30s。

#### dwTimeOutCount

[IN]心跳超时次数。如果设为 0,表示默认次数为 3 次。

#### **byRes**

[IN]保留,设为0。

## 相关接口

NET\_EALARM\_StartListen

## 15. 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

## 结构体定义

## 成员

#### byDeviceID

设备 ID。

#### bySampleTime

GPS 采样时间,格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

#### byTimeZoneIdx

时区。

#### byRetranseFlag

重传标记: 0-实时包, 1-重传包。

#### byRes

保留。

### struGpsInfo

GPS 信息,参见结构体 NET EHOME MPGPS。

#### struMPData

客流统计信息,参见结构体 NET\_EHOME\_MPDATA 。

### 15. 1. 10 NET\_EHOME\_ALARM\_MSG

报警信息结构体

# 结构体定义

```
struct{
 DWORD
           dwAlarmType;
           *pAlarmInfo;
 void
 DWORD
           dwAlarmInfoLen;
           *pXmlBuf;
 void
 DWORD
           dwXmlBufLen;
 char
           sSerialNumber[NET EHOME SERIAL LEN/*12*/];
 void
           *pHttpUr1;
           dwHttpUrlLen;
 DWORD
           byRes[12]:
 BYTE
NET EHOME ALARM MSG, *LPNET EHOME ALARM MSG;
```

### 成员

### dwAlarmType

报警类型。

### pAlarmInfo

报警信息详情,不同报警类型对应不同报警详情。

### dwAlarmInfoLen

保存报警信息详情所需的缓冲区大小。

### pXmlBuf

XML 格式的报警信息报文。

### dwXmlBufLen

保存报警信息 XML 报文所需的缓冲区大小。

### sSerialNumber

设备序列号。

#### pHttpUr1

URL.

 ${\tt dwHttpUrlLen}$ 

URL 大小。

byRes

保留,设为0。

# 备注

# 表 15-1 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	NET_EHOME_ALARM_INFO
EHOME_ALARM_HEATMAP_ REPORT	热度图报警	NET_EHOME_HEATMAP_REPORT
EHOME_ALARM_FACESNAP_ REPORT	人脸抓拍报警	NET_EHOME_FACESNAP_REPORT
EHOME_ALARM_GPS	GPS 报警	NET_EHOME_GPS_INFO
EHOME_ALARM_CID_REPORT	报警主机的 CID 报警	NET_EHOME_CID_INFO
EHOME_ALARM_NOTICE_ PICURL	图片 URL 报警	NET_EHOME_NOTICE_PICURL
EHOME_ALARM_NOTIFY_FAIL	异步失败通知报警	NET_EHOME_NOTIFY_FAIL_INF O
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	NET_EHOME_ALARMWIRELESSIN FO

dwA1armType	描述	pAlarmInfo/pXmlBuf
EHOME_ISAPI_ALARM	通过 HTTP 上传的报警	NET_EHOME_ALARM_ISAPI_INF 0
EHOME_INFO_RELEASE_ PRIVATE	通过 ISUP 上传的报警	NET_EHOME_TERMINAL_REPORT _INFO
EHOME_ALARM_MPDCDATA	车载客流统计报警	NET_EHOME_ALARM_MPDCDATA

# 15. 1. 11 NET\_EHOME\_ALARM\_STATUS\_UNION

报警状态联合体

# 联合体定义

union {	
BYTE	<b>byRes</b> [12];
NET_EHOME_DEV_STATUS_CHANGED	struDevStatusChanged;
NET_EHOME_CHAN_STATUS_CHANGED	struChanStatusChanged;
NET_EHOME_HD_STATUS_CHANGED	struHdStatusChanged;
NET_EHOME_DEV_TIMING_STATUS	struDevTimeStatus;
NET_EHOME_CHAN_TIMING_STATUS_SINGLE	struChanTimeStatus;
NET_EHOME_HD_TIMING_STATUS_SINGLE	struHdTimeStatus;
NET_EHOME_ALARM_STATUS_UNION, *LPNET_E	EHOME_ALARM_STATUS_UNION;

# 成员

### **byRes**

联合体大小。

### struDevStatusChanged

设备状态改变信息,参见结构体 NET\_EHOME\_DEV\_STATUS\_CHANGED 。

### struChanStatusChanged

通道状态改变信息,参见结构体 NET\_EHOME\_CHAN\_STATUS\_CHANGED 。

#### struHdStatusChanged

硬盘状态改变信息,参见结构体 NET\_EHOME\_HD\_STATUS\_CHANGED 。

### struDevTimeStatus

定时上传设备状态参数,参见结构体。 NET\_EHOME\_DEV\_TIMING\_STATUS

#### struChanTimeStatus

定时上传单通道状态参数,参见结构体 NET\_EHOME\_CHAN\_TIMING\_STATUS\_SINGLE 。

#### struHdTimeStatus

定时上传硬盘状态参数,参见结构体 NET\_EHOME\_HD\_TIMING\_STATUS\_SINGLE 。

### 15. 1. 12 NET EHOME ALARM TIME CFG

布防计划参数结构体

### 结构体定义

### 成员

#### dwSize

结构体大小。

#### struSchedTime

布防时间段,参见结构体 **NET\_EHOME\_SCHEDTIME**;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。

#### bySchedTimeCount

布防计划中的时间段个数,只读。

#### **byRes**

保留。

### 15. 1. 13 NET\_EHOME\_ALARM\_TIME\_COND

关于布防计划配置条件参数的结构体。

### 结构体定义

```
struct {
   DWORD     dwSize;
   DWORD     byAlarmType;
   DWORD     byWeekday;
   BYTE     byRes1[2];
   BYTE     dwChanne1;
   BYTE     byRes2[20];
} NET_EHOME_ALARM_TIME_COND, *LPNET_EHOME_ALARM_TIME_COND;
```

### 成员

#### dwSize

结构体大小。

#### byAlarmType

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**

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

保留,设为0。

#### dwChannel

通道号,从1开始。

### byRes2

保留,设为0。

### 15. 1. 14 NET\_EHOME\_ALARMIN\_CFG

报警输入参数结构体。

# 结构体描定义

```
struct{
 DWORD
  DWORD
                                     sAlarmInName[NAME_LEN/*32*/];
                                     byAlarmInType;
 BYTE
                                     byUseAlarmIn;
  BYTE
  BYTE
                                     byRes1[2];
  NET EHOME ALARMIN LINKAGE TYPE
                                     struLinkageType;
 NET EHOME RECORD CHAN
                                     struRecordChan;
                                     byRes2[128];
} NET_EHOME_ALARMIN_CFG, *LPNET_EHOME_ALARMIN_CFG;
```

### 成员

### dwSize

结构体大小。

#### sAlarmInName

报警输入名称。

### byAlarmInType

报警设备状态: 0-常开, 1-常闭。

### byUseAlarmIn

是否开启报警输入: 0-关闭, 1-开启。

### byRes1

保留,设为0。

### struLinkageType

报警联动模式,参见结构体 NET\_EHOME\_ALARMIN\_LINKAGE\_TYPE 。

#### struRecordChan

关联录像通道,参见结构体 NET\_EHOME\_RECORD\_CHAN 。

### byRes1

保留,设为0。

## 15. 1. 15 NET\_EHOME\_ALARMIN\_COND

关于报警输入配置条件参数的结构体。

### 结构体定义

```
struct {
  DWORD    dwSize;
  DWORD    dwAlarmInNum;
  DWORD    dwPTZChan;
  BYTE    byRes[20];
}NET_EHOME_ALARMIN_COND, *LPNET_EHOME_ALARMIN_COND;
```

### 成员

#### dwSize

结构体大小。

### dwAlarmInNum

报警输入号,从1开始。

#### dwPTZChan

PTZ 联动视频通道号,从1开始。

#### **byRes**

保留,设为0。

### 15. 1. 16 NET\_EHOME\_ALARMIN\_LINKAGE\_TYPE

报警联动类型结构体

### 结构体定义

```
struct{
  BYTE
                                byMonitorAlarm;
  BYTE
                                bySoundAlarm;
 BYTE
                                byUpload;
                                byAlarmout;
  BYTE
  BYTE
                                byEmail;
 BYTE
                                byRes1[3];
 NET EHOME LINKAGE PTZ
                                struPTZLinkage;
  NET_EHOME_LINKAGE ALARMOUT
                                struAlarmOut;
                                byRes[128];
NET EHOME ALARMIN LINKAGE TYPE, *LPNET EHOME ALARMIN LINKAGE TYPE;
```

### 成员

### byMonitorAlarm

报警是否上墙: 0-不上墙, 1-上墙。

### bySound

是否开启声音警告: 0-不开启, 1-开启。

### byUpload

是否上传报警至中心: 0-不上传, 1-上传。

#### **byAlarmout**

是否触发报警输出: 0-不触发, 1-触发。

#### byEmail

是否开启邮件联动: 0-不开启, 1-开启。

#### byRes1

保留,设为0。

#### struPTZLinkage

PTZ 联动参数,参见结构体 NET\_EHOME\_LINKAGE\_PTZ。

#### struAlarmOut

报警输出联动参数,参见结构体 NET\_EHOME\_LINKAGE\_ALARMOUT。

### byRes

保留,设为0。

# 15. 1. 17 NET\_EHOME\_ALARMOUT\_CFG

报警输出参数结构体

### 结构体定义

```
struct{
  DWORD    dwSize;
  BYTE    sAlarmOutName[NAME_LEN/*32*/];
  WORD    wAlarmOutDelay;
  BYTE    byRes[26];
} NET_EHOME_ALARMOUT_CFG, *LPNET_EHOME_ALARMOUT_CFG;
```

### 成员

#### dwSize

结构体大小。

#### sAlarmOutName

报警输出名称。

#### byAlarmInType

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

保留,设为0。

# 15. 1. 18 NET\_EHOME\_ALARMOUT\_STATUS\_CFG

报警输出状态结构体

### 结构体定义

```
struct {
  DWORD    dwSize;
  BYTE    byAlarmOutStatus;
  BYTE    byRes[11];
} NET_EHOME_ALARMOUT_STATUS_CFG, *LPNET_EHOME_ALARMOUT_STATUS_CFG;
```

### 成员

### dwSize

结构体大小。

#### byAlarmOutStatus

报警输出状态:1-开启,0-关闭。

### byRes

保留。

## 15. 1. 19 NET\_EHOME\_ALARMWIRELESSINFO

无线报警信息结构体

# 结构体定义

```
struct {
   BYTE     byDeviceID[MAX_DEVICE_ID_LEN/*256*/];
   DWORD    dwDataTraffic;
   BYTE    bySignalIntensity;
   BYTE    byRes[127];
} NET_EHOME_ALARMWIRELESSINFO, *LPNET_EHOME_ALARMWIRELESSINFO;
```

### 成员

byDeviceID

设备 ID。

dwDataTraffic

已用流量,单位:MB。

bySignalIntensity

信号强度,取值范围从0到100。

byRes

保留。

# 15. 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

# 结构体定义

### 成员

#### dwSize

结构体大小。

### byEnable

是否接收报警: 0-否, 1-是。

### byRes1

保留,设为0。

#### struAddress

报警管理服务器的本地回环地址信息(如 IP 地址和端口号),参见结构体 **NET EHOME IPADDRESS** 。

#### byRes2

保留,设为0。

### 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。

# 15. 1. 21 NET\_EHOME\_ASYNC\_RESP\_CB\_DATA

回调响应数据结构体

# 结构体定义

```
struct {
  void*  pOutBuffer;
  DWORD  dwOutLen;
  DWORD  dwErrorNo;
  DWORD  dwHandle;
  LONG  1UserID;
  char  byRes[32];
}NET_EHOME_ASYNC_RESP_CB_DATA, *LPNET_EHOME_ASYNC_RESP_CB_DATA;
```

### 成员

### pOutBuffer

响应数据详情。

#### dw0utLen

响应数据大小。

#### dwErrorNo

错误码。

#### dwHandle

消息句柄,用来匹配请求和响应。

### 1UserID

用户 ID。

### byRes

保留。

# 15. 1. 22 NET\_EHOME\_BINARY\_RECV\_DATA

数据接收参数结构体

### 结构体定义

```
struct {
   DWORD dwSize;
   void *1pOutBuffer;
   DWORD dwOutBufferSize;
   BYTE byRes[64];
} NET_EHOME_BINARY_RECV_DATA, *LPNET_EHOME_BINARY_RECV_DATA;
```

# 成员

### dwSize

结构体大小。

#### 1pOutBuffer

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,详见以下表格。

dwCommand	命令号	1pOutBuffer
NET_EHOME_S2C_TIME_ADJUST	0x9	无
NET_EHOME_S2C_SET_SERVER_INFO	0x18	无
NET_EHOME_S2C_GET_TERMINAL_ INFO	0x19	NET_EHOME_TERMINAL_INFO
NET_EHOME_S2C_POST_SCHEDULE	0x50	无

# ISUPSDK(通用) 开发指南

dwCommand	命令号	1pOutBuffer
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	NET_EHOME_DEV_WORK_STATUS
NET_EHOME_S2C_VERSIONUPGRADE	0x57	无
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	NET_EHOME_DEV_DEFAULT_PARAM
NET_EHOME_S2C_SET_QUEUE_INFO	0x66	无
NET_EHOME_S2C_PUBLISH_SHOT_PIC	0x67	无
NET_EHOME_S2C_GET_IPCINFO	0x68	NET_EHOME_RELATE_IPC_INFO
NET_EHOME_S2C_SET_TERMINAL_IP	0x69	无
NET_EHOME_S2C_GET_TERMINAL_IP	0x70	NET_EHOME_TERM_IP_ADDRESS
NET_EHOME_S2C_SET_SWITCHPLAN	0x100	无
NET_EHOME_S2C_GET_SWITCHPLAN	0x101	NET_EHOME_SWITCH_PLAN_PARAM
NET_EHOME_S2C_SET_VOLUMEPLAN	0x102	无
NET_EHOME_S2C_GET_VOLUMEPLAN	0x103	NET_EHOME_VOLUME_PLAN_PARAM
NET_EHOME_S2C_SET_CHARACTER	0x106	无
NET_EHOME_S2C_GET_CHARACTER	0x107	NET_EHOME_INSERT_CHARACTER

dwCommand	命令号	1pOutBuffer
NET_EHOME_S2C_SET_ADB_CFG	0x110	无
NET_EHOME_S2C_GET_ADB_CFG	0x111	NET_EHOME_ADB_DEBUG
NET_EHOME_S2C_SET_TIEM_ZONE	0x112	无
NET_EHOME_S2C_GET_TIME_ZONE	0x113	NET_EHOME_TERM_TIME_ZONE
NET_EHOME_S2C_SET_SADP	0x114	无
NET_EHOME_S2C_GET_SADP	0x115	NET_EHOME_DISCOVERY_MODE
NET_EHOME_S2C_GET_COMPONENT	0x116	NET_EHOME_COMPONENT_GROUP_INFO
NET_EHOME_S2C_TRANS_WITH_RET	0x10A	无
NET_EHOME_S2C_PUBLISH_XML	0x200	无
NET_EHOME_S2C_SCREEN_SHOT	0x201	NET_EHOME_SCREEN_SHOT_RET

### dwOutBufferSize

已接收的数据大小。

### byRes

保留。

# 15. 1. 23 NET\_EHOME\_BINARY\_SEND\_DATA

数据发送参数结构体。

# 结构体定义

```
struct{
  DWORD dwSize;
  void *1pInBuffer;
  DWORD dwInBufferSize;
  DWORD dwRecvTimeOut;
  BYTE byRes[64];
}NET_EHOME_BINARY_SEND_DATA, *LPNET_EHOME_BINARY_SEND_DATA;
```

# 成员

### dwSize

结构体大小。

# 1pInBuffer

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,详见以下表格。

dwCommand	命令号	lpInBuffer
NET_EHOME_S2C_TIME_ADJUST	0x9	NET_EHOME_TIME_ADJUST_PARAM
NET_EHOME_S2C_SEND_ERRCODE	0xD	NET_EHOME_ERRCODE_RET
NET_EHOME_S2C_SET_SERVER_INFO	0x18	NET_EHOME_RELEASE_SERVER_INFO
NET_EHOME_S2C_GET_TERMINAL_ INFO	0x19	无
NET_EHOME_S2C_LOGOUT	0x20	无
NET_EHOME_S2C_POST_SCHEDULE	0x50	NET_EHOME_POST_SCHEDULE
NET_EHOME_S2C_TERMINALCONTROL_ V20	0x51	NET_EHOME_TERMINAL_CONTROL_V20
NET_EHOME_S2C_TERMINALCONTROL	0x53	NET_EHOME_TERMINAL_CONTROL
NET_EHOME_S2C_PLAYCONTROL	0x54	NET_EHOME_TERMINAL_CONTROL
NET_EHOME_S2C_SET_IPCINFO	0x55	NET_EHOME_RELATE_IPC_INFO
NET_EHOME_S2C_GET_WORKSTATUS	0x56	无
NET_EHOME_S2C_VERSIONUPGRADE	0x57	NET_EHOME_UPGRADE_CONN_PARAM
NET_EHOME_S2C_SETPLAYERPARAM	0x58	NET_EHOME_PLAYER_PARAM
NET_EHOME_S2C_SETSERVERADDR	0x5B	NET_EHOME_PUBL   SH_SERVERADDR
NET_EHOME_WEATHERINFO	0x61	NET_EHOME_WEATHER_INFO
NET_EHOME_S2C_GET_SCREENSHOT	0x62	NET_EHOME_SCREEN_SHOT
NET_EHOME_S2C_GET_DEFAULTPARAM	0x63	无
NET_EHOME_S2C_REPLACE_MATERIAL	0x64	NET_EHOME_REPLACE_MATERIAL
NET_EHOME_S2C_CANCEL_SCHEDULE	0x65	NET_EHOME_POST_SCHEDULE
NET_EHOME_S2C_SET_QUEUE_INFO	0x66	NET_EHOME_QUEUE_INFO

# ISUPSDK(通用) 开发指南

dwCommand	命令号	lpInBuffer
NET_EHOME_S2C_PUBLISH_SHOT_PIC	0x67	NET_EHOME_SHOT_PIC
NET_EHOME_S2C_GET_IPCINFO	0x68	无
NET_EHOME_S2C_SET_TERMINAL_IP	0x69	NET_EHOME_TERM_IP_ADDRESS
NET_EHOME_S2C_GET_TERMINAL_IP	0x70	无
NET_EHOME_S2C_SET_TERMINAL_ NAME	0x71	NET_EHOME_TERMINAL_NAME
NET_EHOME_S2C_SET_SWITCHPLAN	0x100	NET_EHOME_SWITCH_PLAN_PARAM
NET_EHOME_S2C_GET_SWITCHPLAN	0x101	无
NET_EHOME_S2C_SET_VOLUMEPLAN	0x102	NET_EHOME_VOLUME_PLAN_PARAM
NET_EHOME_S2C_GET_VOLUMEPLAN	0x103	无
NET_EHOME_S2C_SET_CHARACTER	0x106	NET_EHOME_INSERT_CHARACTER
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	NET_EHOME_ADB_DEBUG
NET_EHOME_S2C_GET_ADB_CFG	0x111	无
NET_EHOME_S2C_SET_TIEM_ZONE	0x112	NET_EHOME_TERM_TIME_ZONE
NET_EHOME_S2C_GET_TIME_ZONE	0x113	无
NET_EHOME_S2C_SET_SADP	0x114	NET_EHOME_DISCOVERY_MODE
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	NET_EHOME_POST_PUBL   SH_XML
NET_EHOME_S2C_SCREEN_SHOT	0x201	NET_EHOME_SCREEN_SHOT_EX

dw In Buffer Size

待发送数据大小。

#### dwRecvTimeOut

接收超时时间。

### byRes

保留。

# 15. 1. 24 NET\_EHOME\_BLACKLIST\_SEVER

黑名单管理服务器参数结构体

### 结构体定义

### 成员

struAdd

服务器地址,参见 NET\_EHOME\_IPADDRESS 。

byServerName

服务器名称。

byUserName

用户名。

byPassWord

密码。

# 15. 1. 25 NET\_EHOME\_CHAN\_STATUS\_CHANGED

通道状态改变参数结构体

### 结构体定义

```
struct{
WORD wChanNO;
BYTE byChanStatus;
```

```
BYTE byRes[9];
```

NET EHOME CHAN STATUS CHANGED, \*LPNET EHOME CHAN STATUS CHANGED;

### 成员

#### wChanNO

通道号

### byChanStatus

通道状态, 按位表示。

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

保留。

### 15. 1. 26 NET\_EHOME\_CHAN\_TIMING\_STATUS\_SINGLE

单通道状态定时上报参数结构体

# 结构体定义

```
struct {
   DWORD dwBitRate;
   WORD wChanNO;
   BYTE byLinkNum;
   BYTE byRes[5];
} NET_EHOME_CHAN_TIMING_STATUS_SINGLE, *LPNET_EHOME_CHAN_TIMING_STATUS_SINGLE;
```

### 成员

#### dwBitRate

实际码率,单位: Kbps。

#### wChanNO

通道号。

### byLinkNum

关联的客户端个数。

### byRes

保留。

# 15. 1. 27 NET\_EHOME\_CHAR\_EFFECT

字符效果结构体

### 结构体定义

```
struct{
 DWORD
          dwFontSize;
 DWORD
          dwFontColor;
 DWORD
          dwBackColor:
          dwBackTransparent;
 DWORD
          bSubtitlesEnabled;
 BOOL
 BYTE
          szScrollDirection[32];
 DWORD
         dwScrol1Speed;
}NET_EHOME_CHAR_EFFECT, *LPNET_EHOME_CHAR_EFFECT;
```

# 成员

dwFontSize

字体大小。

dwFontColor

字体颜色。

dwBackColor

背景颜色。

dwBackTransparent

背景透明度。

bSubtitlesEnabled

是否启用 OSD。

szScrollDirection

滚动方向: 向左, 向右, 向上, 向下。

dwScrollSpeed

滚动速度。

# 15. 1. 28 NET\_EHOME\_CID\_INFO

CID 信息结构体

### 结构体定义

```
struct{
  DWORD
                         dwSize:
                         byDeviceID[MAX DEVICE ID LEN/*256*/];
  char
  DWORD
                         dwCIDCode;
  DWORD
                         dwCIDType;
  DWORD
                         dwSubSysNo;
                         byCIDDescribe[CID_DES_LEN/*32*/];
  char
                         byTriggerTime[MAX_TIME_LEN/*32*/];
  char
  char
                         byUploadTime[MAX_TIME_LEN/*32*/];
  NET_EHOME_CID_PARAM
                         struCIDParam;
  BYTE
                    byTimeDiffH;
  BYTE
                    byTimeDiffM;
  BYTE
                         byExtend;
                         byRes1[5];
  BYTE
  void
                         *pCidInfoEx;
  BYTE
                         byRes[52];
}NET_EHOME_CID_INFO, *LPNET_EHOME_CID_INFO;
```

### 成员

#### dwSize

结构体大小。

#### byDeviceID

设备注册 ID。

#### dwCIDCode

CID码。

#### dwCIDType

CID 报警类型。

#### dwSubSysNo

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

CID 报警类型。

#### byTriggerTime

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)。byUploadTime

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)。struCIDParam

CID 报警参数, 详见 NET\_EHOME\_CID\_PARAM。

### byTimeDiffH

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

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

是否存在扩展字段: 0-否, 1-是。

#### bvRes1

保留。

### pCidInfoEx

只有当 byExtend 设为 1 时,该参数有效,并指向结构体 NET\_EHOME\_CID\_INFO\_INTERNAL\_EX

### byRes

保留。

### 15. 1. 29 NET\_EHOME\_CID\_INFO\_INTERNAL\_EX

CID报警扩展信息结构体

# 结构体定义

```
struct{
   BYTE byRecheck;
BYTE byRes[3];
char byUUID[MAX_UUID_LEN/*64*/];
char byVideoURL[MAX_URL_LEN/*512*/];
char byCIDDescribeEx[CID_DES_LEN_EX/*256*/];
BYTE byRes1[256];
}NET_EHOME_CID_INFO_INTERNAL_EX, *LPNET_EHOME_CID_INFO_INTERNAL_EX;
```

### 成员

### byRecheck

报警类型:1-视频复核报警,0-普通报警。

### byRes

保留。

### byUUID

报警 ID。当报警类型为视频复核报警时,该参数应设为 0.

### byVideoURL

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。

### byCIDDescribeEx

CID报警扩展信息。

### byRes1

保留。

### 15. 1. 30 NET\_EHOME\_CID\_PARAM

CID 报警信息结构体

### 结构体定义

```
struct{
 DWORD
           dwUserType;
           1UserNo;
  LONG
  LONG
           1ZoneNo:
           1KeyboardNo;
  LONG
 LONG
           lVideoChanNo;
           lDiskNo:
 LONG
           1ModuleAddr;
 LONG
           byUserName[NAME LEN/*32*/];
  char
           byRes[32];
 BYTE
} NET_EHOME_CID_PARAM, *LPNET_EHOME_CID_PARAM;
```

### 成员

### dwUserType

用户类型: 1-键盘用户, 2-网络用户, 其他值表示无效。

#### 1UserNo

用户编号,当该参数值设为-1时,表示该参数无效。

#### 1ZoneNo

防区号, 当该参数值设为-1时, 表示该参数无效。

#### 1KeyboardNo

键盘编号。

#### 1VideoChanNo

视频通道号。

#### 1DiskNo

硬盘号。

### 1ModuleAddr

模块地址。

### byUserName

用户名。

#### **byRes**

保留。

# 15. 1. 31 NET\_EHOME\_CMS\_LISTEN\_PARAM

中心管理服务器监听参数结构体

### 结构体定义

### 成员

### struAddress

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 **NET\_EHOME\_IPADDRESS**。

#### fnCB

注册回调函数,参见 DEVICE\_REGISTER\_CB。

### pUserData

用户数据。

#### **byRes**

保留,设为0。

### 相关接口

NET\_ECMS\_StartListen

### 15. 1. 32 NET\_EHOME\_CMSCB\_DATA

回调数据结构体

# 结构体定义

```
struct{
  LONG lUserID;
  DWORD dwType;
  DWORD dwHandle;
  BOOL bSucc;
  void* pOutBuffer;
  DWORD dwOutLen;
  DWORD dwErrorNo;
  BYTE byRes[32];
}NET_EHOME_CMSCB_DATA, *LPNET_EHOME_CMSCB_DATA;
```

# 成员

1UserID

用户 ID。

### dwType

回调的数据类型,参见以下表格。

类型宏定义	宏定义值	描述
ECMS_CB_TYPE_UNDEFINE	0	未定义
ECMS_CB_TYPE_START_GET_ REALSTREAM	1	开始预览的请求数据
ECMS_CB_TYPE_START_PUSH_ REALSTREAM	3	开始传输实时流的请求数据
ECMS_CB_TYPE_STOP_GET_ REALSTREAM_EX	4	停止预览的请求数据
ECMS_CB_TYPE_START_PLAYBACK	5	开始回放的请求数据
ECMS_CB_TYPE_START_PUSH_ PLAYBACK	6	开始传输回放码流的请求数据
ECMS_CB_TYPE_STOP_PLAYBACK_ EX	7	停止回放的请求数据
ECMS_CB_STOP_VOICETALK_EX	9	停止语音对讲的请求数据
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

类型宏定义	宏定义值	描述
ECMS_CB_TYPE_START_PUSH_ VOICE_STREAM	11	开始传输语音码流的请求数据
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
ECMS_CB_TYPE_PAUSE_PLAYBACK	13	暂停回放的请求数据
ECMS_CB_TYPE_RESTART_ PLAYBACK	14	继续回放的请求数据

### dwHandle

消息句柄。

#### bSucc

同步返回接口数据。

### pOutBuffer

设备响应数据。

#### dwOutLen

设备响应数据大小。

### dwErrorNo

错误码。

### byRes

保留。

# 相关接口

NET\_ECMS\_SetCallback

# 15. 1. 33 NET\_EHOME\_COMPONENT\_GROUP\_INFO

模块分组信息结构体

# 结构体定义

```
BYTE byRes[20];
}NET_EHOME_COMPONENT_GROUP_INFO, *LPNET_EHOME_COMPONENT_GROUP_INFO;
```

### 成员

### dwComponentNum

模块个数。

### struSysComponentAddr

模块信息,详见 NET\_EHOME\_SYSCOMPONENT\_REDUCED\_ADDR。

### **byRes**

保留。

### 15. 1. 34 NET\_EHOME\_COMPRESSION\_CFG

编码参数结构体

### 结构体定义

```
struct{
 DWORD
           dwSize;
  BYTE
           byStreamType:
           byPicQuality;
  BYTE
  BYTE
           byBitRateType;
           byRes1;
  BYTE
  DWORD
           dwResolution:
  DWORD
           dwVideoBitRate:
           dwMaxBitRate;
  DWORD
           dwVideoFrameRate;
  DWORD
  WORD
           wIntervalFrameI:
  BYTE
           byIntervalBPFrame;
           byRes[41];
 BYTE
}NET EHOME COMPRESSION_CFG, *LPNET_EHOME_COMPRESSION_CFG;
```

### 成员

### dwSize

结构体大小。

### byStreamType

码流类型: 0-视频流, 1-视频和音频流。

#### byPicQuality

图像质量: 0-最高, 1-较高, 2-中等, 3-低, 4-较低, 5-最低。

#### byBitRateType

码率类型: 0-变码率, 1-定码率。

#### byRes1

保留,设为0。

#### dwResolution

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

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 Kbps, 20-1792 Kbps, 21-2048 Kbps, 22-custom。

#### dwMaxBitRate

自定义码率,只有当 dwVideoBitRate 设为 22 时有效。

#### dwVideoFrameRate

视频帧率: 0-所有帧,1-1/16×,2-1/8×,3-1/4×,4-1/2×,5-1×,6-2×,7-4×,8-6×,9-8×,10-10×,11-12×,12-16×,13-20×,14-15×,15-18×,16-22×。

#### wIntervalFrameI

I 帧间隔,范围从1到400。

#### byIntervalBPFrame

帧类型: 0-BBP 帧, 1-BP 帧, 2-P 帧。

#### **byRes**

保留,设为0。

# 15. 1. 35 NET\_EHOME\_COMPRESSION\_COND

编码设置条件参数结构体

### 结构体定义

```
struct{
   DWORD    dwSize;
   DWORD    dwChannelNum;

BYTE    byCompressionType;

BYTE    byRes[23];
} NET_EHOME_COMPRESSION_COND, *LPNET_EHOME_COMPRESSION_COND;
```

### 成员

dwSize

结构体大小。

dwChannelNum

通道号,从1开始。

byCompressionType

码流类型:1-主码流,2-子码流,3-第三码流。

**byRes** 

保留,设为0。

### 15. 1. 36 NET\_EHOME\_CONFIG

配置参数结构体

# 结构体定义

```
struct{
  void
           *pCondBuf;
  DWORD
           dwCondSize;
           *pInBuf:
  void
  DWORD
           dwInSize;
           *pOutBuf;
  void
  DWORD
           dwOutSize;
  BYTE
           byRes[40];
}NET_EHOME_CONFIG, *LPNET_EHOME_CONFIG;
```

### 成员

pCondBuf

条件参数,如通道号;获取参数时,该参数应设为"NULL"。

#### dwCondSize

条件缓冲区大小。

### pInBuf

输入参数, 获取配置参数时, 该参数应设为"NULL"。

#### dwInSize

输入缓冲区大小。

#### pOutBuf

输出参数,设置参数时,该参数应设为"NULL"。

### dwOutSize

输出缓冲区大小。

#### **byRes**

保留,设为0。

# 15. 1. 37 NET\_EHOME\_DEV\_DEFAULT\_PARAM

终端设备默认参数结构体。

## 结构体定义

```
struct {
 BOOL
          bAutoLightEnable;
 DWORD
          dwLightMode;
 DWORD
          dwBackLightLevel;
          dwVolume;
 DWORD
          bEnableStartLogo;
 BOOL
 BOOL
         bEnableVolumeTiming;
 BOOL
          bEnablePowerTiming;
 BOOL
          bDefaultScheduleEnable:
          dwDefaultScheduleId:
 DWORD
 BYTE
          szDefaultScheduleName[32];
          bTemperaturenable;
 BOOL
 DWORD
          dwSecurity;
 DWORD
          dwProtectValue;
}NET_EHOME_DEV_DEFAULT_PARAM, *LPNET_EHOME_DEV_DEFAULT_PARAM;
```

### 成员

#### bAutoLightEnable

是否启用自动亮度调节。

### dwLightMode

亮度值。

### dwBackLightLevel

背光参数。

#### dwVolume

音量。

### bEnableStartLogo

是否显示开机 LOGO。

### bEnableVolumeTiming

是否启用音量定时调节计划。

### bEnablePowerTiming

是否启用定时开关机计划。

### bDefaultScheduleEnable

是否启用垫片计划。

#### dwDefaultScheduleId

垫片计划 ID。

### szDefaultScheduleName

垫片计划名称。

#### bTemperaturenable

是否启用安全温度功能。

### dwSecurity

温度的安全值。

#### dwProtectValue

温度的保护值。

# 15. 1. 38 NET\_EHOME\_DEV\_LOG

关于终端日志查询结果的结构体。

### 结构体定义

struct{

NET\_EHOME\_TIME struLogTime;
DWORD dwMajorType;
DWORD dwMinorType;

```
DWORD
                     dwParamType;
                     sLocalUser[NAME LEN/*32*/];
  char
                     sRemoteUser[NAME LEN/*32*/];
  char
                     sIPAddress[128];
  char
                     dwChannelNo;
  DWORD
  DWORD
                     dwHardDiskNo;
                     dwAlarmInputChanNo;
  DWORD
  DWORD
                     dwAlarmOutputChanNo;
                     sLogContext[MAX LOG INFO LEN/*8 × 1024*/];
  char
                    byRes[64];
} NET EHOME DEV LOG, *LPNET EHOME DEV LOG;
```

### 成员

#### struLogTime

日志时间,参见 NET\_EHOME\_TIME 。

#### dwMajorType

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

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

日志次类型参数,参见 **设备日志类型**。

#### sLocalUser

本地用户名。

#### sRemoteUser

远程用户名。

#### sIPAddress

远程主机 IP 地址。

#### dwChannelNo

通道号。

#### dwHardDiskNo

硬盘号。

### dwAlarmInputChanNo

报警输入通道号。

### dwAlarmOutputChanNo

报警输出通道号。

### sLogContext

日志信息。

### byRes

保留,设为0。

# 15. 1. 39 NET\_EHOME\_DEV\_LOG\_COND

关于设备日志查询条件的结构体。

### 结构体定义

```
struct{
 DWORD
                    dwMajorType;
  DWORD
                    dwMinorType;
  NET_EHOME_TIME
                    struStartTime;
  NET_EHOME_TIME
                    struStopTime;
  DWORD
                    dwStartIndex;
  DWORD
                    dwMaxFileCountPer;
 BYTE
                    byLocalOrUTC;
                    byRes[63];
  BYTE
NET EHOME DEV LOG COND, *LPNET EHOME DEV LOG COND;
```

### 成员

### dwMajorType

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

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

开始时间,可参见 NET EHOME TIME。

#### struStopTime

结束时间,可参见 NET EHOME TIME 。

#### dwStartIndex

查询起始位置,从0开始。

#### dwMaxFileCountPer

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

时间类型: 0-设备本地时间, 即设备 OSD 时间: 1-UTC 时间。

### byRes

保留,设为0。

# 15. 1. 40 NET\_EHOME\_DEV\_REG\_INFO

设备注册信息结构体。

### 结构体定义

```
struct{
 DWORD
                       dwSize;
 DWORD
                       dwNetUnitType;
  BYTE
                       byDeviceID[MAX_DEVICE_ID_LEN/*256*/];
  BYTE
                       byFirmwareVersion[24];
  NET EHOME IPADDRESS
                       struDevAdd;
  DWORD
                       dwDevType;
  DWORD
                       dwManufacture;
  BYTE
                       byPassWord[32];
                       sDeviceSerial[NET EHOME SERIAL LEN/*12*/];
  BYTE
  BYTE
                       byReliableTransmission;
  BYTE
                       byWebSocketTransmission;
  BYTE
                       bySupportRedirect;
                       byDevProtocolVersion[6];
  BYTE
  BYTE
                       bySessionKey[MAX_MASTER_KEY_LEN/*16*/];
 BYTE
                       byRes[27];
} NET EHOME DEV REG INFO, *LPNET EHOME DEV REG INFO;
```

### 成员

### dwSize

结构体大小。

### dwNetUnitType

保留。

#### byDeviceID

设备 ID。

### byFirmwareVersion

固件版本信息。

#### struDevAdd

设备地址,参见结构体 NET\_EHOME\_IPADDRESS。

### dwDevType

设备类型。

#### dwManufacture

制造商代码。

### byPassWord

设备登录中心管理服务器的密码,用户根据需求进行验证。

#### sDeviceSerial

设备序列号。

### byReliableTransmission

可靠传输。

### byWebSocketTransmission

网络接口传输。

### bySupportRedirect

是否支持重定向: 0-否, 1-是。

### byDevProtocolVersion

设备协议版本信息。

### bySessionKey

支持 5.0 版本 ISUP 的设备会话密钥。

### byRes

保留。

# 15. 1. 41 NET\_EHOME\_DEV\_REG\_INFO\_V12

设备注册信息结构体(V12)

### 结构体定义

### 成员

### struRegInfo

注册信息,参见 NET\_EHOME\_DEV\_REG\_INFO。

#### struRegAddr

设备注册的服务器地址,参见结构体 NET\_EHOME\_IPADDRESS。

#### sDevName

设备名称。

#### byDeviceFullSerial

设备序列号,包含设备型号,时间,编号和扩展信息。

#### **byRes**

保留。

### 15. 1. 42 NET EHOME DEV SESSIONKEY

支持 5.0 版本 ISUP 的设备会话密钥信息结构体。

### 结构体定义

### 成员

#### sDeviceID

设备 ID。

### sSessionKey

设备的会话密钥。

# 15. 1. 43 NET EHOME DEV SESSIONKEY EHOME50

支持 5.0 版本 ISUP 的设备会话密钥参数结构体

# 结构体定义

```
struct{
BYTE sDeviceID[MAX_DEVICE_ID_LEN/*256*/];
BYTE sSessionKey[MAX_EHOME50_MASTER_KEY_LEN/*16*/];
}NET_EHOME_DEV_SESSIONKEY_EHOME50, *LPNET_EHOME_DEV_SESSIONKEY_EHOME50;
```

### 成员

sDeviceID

设备 ID。

### sSessionKey

设备会话密钥。

## 15. 1. 44 NET\_EHOME\_DEV\_STATUS\_CHANGED

设备状态改变信息结构体

### 结构体定义

```
struct{
  BYTE byDeviceStatus;
  BYTE byRes[11];
}NET_EHOME_DEV_STATUS_CHANGED, *LPNET_EHOME_DEV_STATUS_CHANGED;
```

## 成员

### byDeviceStatus

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**

保留。

# 15. 1. 45 NET\_EHOME\_DEV\_TIMING\_STATUS

关于设备状态定时上传参数的结构体。

# 结构体定义

```
struct {
  DWORD          dwMemoryTotal;
  DWORD          dwMemoryUsage;
  BYTE          byCPUUsage;
  BYTE          byMainFrameTemp;
  BYTE          byBackPanelTemp;
  BYTE          byRes;
}NET_EHOME_DEV_TIMING_STATUS, *LPNET_EHOME_DEV_TIMING_STATUS;
```

### 成员

### dwMemoryTotal

内存总大小,单位: KB。

### dwMemoryUsage

内存使用量,单位: KB。

### byCPUUsage

CPU 使用率 (%), 范围从 0 到 100。

### byMainFrameTemp

机箱温度,单位: °C。

### byBackPanelTemp

背板温度,单位: °C。

#### **byRes**

保留。

### 15. 1. 46 NET EHOME DEV WORK STATUS

设备工作状态结构体。

# 结构体定义

```
struct{
            byIdentifyCode[32];
   BYTE
   DWORD
            dwCpuUsage;
   DWORD
            dwMemTotal;
   DWORD
            dwMemUsed;
   DWORD
            dwDiskTotal;
   DWORD
            dwDiskUsed;
   DWORD
            dwTemperature;
            bySoftwareVersion[32];
   BYTE
            byDspwareVersion[32];
   BYTE
   BYTE
            byHardwareVersion[32];
            bySystemVersion[32];
   BYTE
            byMacAddress[2][32];
   BYTE
            bySwitchEnable;
   BOOL
            byVolumeEnable;
   BOOL
   DWORD
            dwRelateScheNo;
   BYTE
            byScheName[100];
            byInstal1Type[32];
   BYTE
   DWORD
            byResWidth;
            byResHeight:
   DWORD
}NET EHOME DEV WORK STATUS, *LPNET EHOME DEV WORK STATUS;
```

### 成员

### byIdentifyCode

终端序列号。

#### dwCpuUsage

CPU 使用率。

### dwMemTotal

内存总大小。

### dwMemUsed

已使用内存。

### dwDiskTotal

磁盘总容量。

### dwDiskUsed

磁盘利用率。

## dwTemperature

终端温度。

## bySoftwareVersion

软件版本号。

## byDspwareVersion

DSP 版本号。

## byHardwareVersion

硬件版本号。

## bySystemVersion

系统版本号。

### byMacAddress

物理地址,最大支持2个网卡。

### bySwitchEnable

是否启用定时开关机计划。

### byVolumeEnable

是否启用音量定时调节计划。

### dwRelateScheNo

计划编号。

### byScheName

终端显示的计划编号。

### byInstallType

屏幕方向: 竖屏或横屏。

### byResWidth

分辨率 (宽度)。

## byResHeight

分辨率 (高度)。

# 15. 1. 47 NET\_EHOME\_DEVICE\_CFG

设备基本信息结构体。

## 结构体定义

```
struct{
 DWORD
           dwSize;
           sServerName[MAX DEVICE NAME LEN/*32*/];
 BYTE
  DWORD
           dwServerID:
  DWORD
           dwRecycleRecord;
 DWORD
           dwServerType;
           dwChannelNum:
  DWORD
  DWORD
           dwHardDiskNum;
  DWORD
           dwAlarmInNum;
 DWORD
           dwAlarmOutNum;
 DWORD
           dwRS232Num:
 DWORD
           dwRS485Num:
 DWORD
           dwNetworkPortNum;
 DWORD
           dwAuxoutNum;
  DWORD
           dwAudioNum;
  BYTE
           sSerialNumber[MAX SERIALNO LEN/*128*/];
  DWORD
           dwMajorScale;
 DWORD
           dwMinorScale:
  BYTE
           byRes[292];
}NET EHOME DEVICE CFG, *LPNET EHOME DEVICE CFG;
```

## 成员

### dwSize

结构体大小。

### sServerName

设备名称。

#### dwServerID

设备或遥控器编号,范围从0到255。

## dwRecycleRecord

是否支持循环录像: 0-否, 1-是。

### dwServerType

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

通道个数(包括模拟和数字通道),只读。

### dwHardDiskNum

硬盘个数,只读。

#### dwAlarmInNum

模拟通道关联的报警输入个数,只读。

### dwAlarmOutNum

模拟通道关联的报警输出个数,只读。

### dwRS232Num

RS-232 串口个数,只读。

### dwRS485Num

RS-485 串口个数,只读。

### dwNetworkPortNum

网口个数, 只读。

### dwAuxoutNum

辅口个数,只读。

### dwAudioNum

音频端口个数,只读。

### sSerialNumber

设备序列号,只读。

### dwMajorScale

是否支持主口缩放: 0-否, 1-是。

### dwMinorScale

是否支持辅口缩放: 0-否, 1-是。

#### **byRes**

保留。

## 15. 1. 48 NET\_EHOME\_DEVICE\_INFO

设备信息结构体。

```
struct{
           dwSize:
  DWORD
  DWORD
           dwChannelNumber:
  DWORD
           dwChannelAmount;
  DWORD
           dwDevType;
  DWORD
           dwDiskNumber;
           sSerialNumber[MAX_SERIALNO_LEN/*128*/];
  BYTE
  DWORD
           dwAlarmInPortNum;
  DWORD
           dwAlarmInAmount;
  DWORD
           dwAlarmOutPortNum;
  DWORD
           dwAlarmOutAmount;
           dwStartChannel;
  DWORD
  DWORD
           dwAudioChanNum;
  DWORD
           dwMaxDigitChannelNum;
  DWORD
           dwAudioEncType;
           sSIMCardSN[MAX SERIALNO LEN/*128*/];
  BYTE
  BYTE
           sSIMCardPhoneNum[MAX PHOMENUM LEN/*32*/];
  DWORD
           dwSupportZeroChan;
           dwStartZeroChan;
  DWORD
           dwSmartType;
  DWORD
  WORD
           wDevClass;
           byRes[158];
  BYTE
}NET EHOME DEVICE INFO, *LPNET EHOME DEVICE INFO;
```

## 成员

### dwSize

结构体大小。

### dwChannelNumber

模拟通道个数。

### dwChannelAmount

通道总数(包括模拟通道和数字通道)。

## dwDevType

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

设备中的硬盘数量。

### sSerialNumber

设备序列号。

### dwAlarmInPortNum

模拟通道关联的报警输入个数。

#### dwAlarmInAmount

报警输入总数。

### dwAlarmOutPortNum

模拟通道关联的报警输出个数。

### dwAlarmOutAmount

报警输出总数。

### dwStartChannel

起始视频通道号。

### dwAudioChanNum

语音对讲通道个数。

## dwMaxDigitChannelNum

设备支持的最大数字通道个数

## dwAudioEncType

语音对讲的音频格式: 0-OggVorbis, 1-G711U, 2-G711A, 3-G726, 4-AAC, 5-MP2L2。

#### sSIMCardSN

SIM卡序列号(车载设备扩展)。

### sSIMCardPhoneNum

SIM卡手机号码(车载设备扩展)。

#### dwSupportZeroChan

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

零通道的起始编号,默认为10000。

### dwSmartType

0-智能(默认),1-专业智能。

#### wDevClass

设备类型。

### **byRes**

保留。

# 15. 1. 49 NET\_EHOME\_DISCOVERY\_MODE

终端 SADP 参数结构体

```
struct{
  BOOL bSadpMode;
}NET_EHOME_DISCOVERY_MODE, *LPNET_EHOME_DISCOVERY_MODE;
```

## 成员

bSadpMode

是否开启 SADP。

# 15. 1. 50 NET\_EHOME\_DISPLAY\_PARAM

终端显示参数结构体

# 结构体定义

```
struct{
    DWORD
            dwR;
    DWORD
            dwG;
    DWORD
            dwB;
    DWORD
            dwOffsetR;
    DWORD
            dwOffsetG:
            dwOffsetB;
    DWORD
    DWORD
            dwBrightnessLevel;
            dwContrastLevel;
    DWORD
    DWORD
            dwSharpnessLevel;
            dwSaturationLevel;
    DWORD
    DWORD
            dwHueLevel;
            bAutoLightEnable;
    BOOL
    DWORD
            dwLightMode;
    DWORD
            dwBackLightLevel;
    BOOL
            bBootLogoEnabled;
    DWORD
            dwScreenVolume:
    BOOL
            bDefaultScheduleEnable;
    DWORD
            dwDefaultScheduleId;
            byDefaultScheduleName[32];
    BYTE
    BOOL
            bTemperaturenable;
    DWORD
            bSecurity;
    DWORD
            bProtectValue;
}NET_EHOME_DISPLAY_PARAM, *LPNET_EHOME_DISPLAY_PARAM;
```

# 成员

dwR

红色。

dwG

绿色。

### dwB

蓝色。

### dw0ffsetR

红色色差量。

### dw0ffsetG

绿色色差量。

## dw0ffsetB

蓝色色差量。

## dwBrightnessLevel

亮度。

### dwContrastLevel

对比度。

# ${\tt dwSharpnessLevel}$

锐度。

### dwSaturationLevel

饱和度。

## dwHueLevel

色调。

### bAutoLightEnable

是否开启亮度自动调节。

## dwLightMode

自动调节的亮度值。

## dwBackLightLevel

背光参数。

### bBootLogoEnabled

是否显示开机 LOGO。

### dwScreenVolume

屏幕音量, 暂时不支持。

### bDefaultScheduleEnable

是否启用垫片计划。

### dwDefaultScheduleId

垫片计划 ID。

### byDefaultScheduleName

垫片计划名称。

### bTemperaturenable

是否启用安全温度功能。

## bSecurity

温度的安全值。

### bProtectValue

温度的保护值。

# 15. 1. 51 NET\_EHOME\_ERRCODE\_RET

错误码返回结构体

# 结构体定义

```
struct{
  DWORD dwCmd;
  DWORD dwErrCode;
}NET_EHOME_ERRCODE_RET, *LPNET_EHOME_ERRCODE_RET;
```

## 成员

### dwCmd

错误码。

### dwErrCode

返回错误码的对应名称,参见 NET\_EHOME\_ERROR\_CODE。

# 15. 1. 52 NET\_EHOME\_ETHERNET

以太网端口信息结构体

## 结构体定义

# ISUPSDK (通用) 开发指南

```
BYTE byMACAddr[MACADDR_LEN/*6*/];
BYTE byRes[2];
}NET_EHOME_ETHERNET, *LPNET_EHOME_ETHERNET;
```

## 成员

#### struDevIP

设备 IP 地址,参见 NET EHOME IPADDR。

#### struDevIPMask

掩码地址,参见 NET\_EHOME\_IPADDR。

### dwNetInterface

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

设备端口号, SDK 的默认通信端口号为8000。

#### wMTU

MTU 参数。

### byMACAddr

设备物理地址。

# 15. 1. 53 NET\_EHOME\_FACESNAP\_REPORT

人脸抓拍报警信息结构体

## 结构体定义

```
struct{
  DWORD
                              dwSize;
                              byDeviceID[MAX DEVICE ID LEN/*256*/];
  char
  DWORD
                              dwVideoChannel;
                              byAlarmTime[MAX_TIME_LEN/*32*/];
  char
  DWORD
                              dwFacePicID;
  DWORD
                              dwFaceScore;
                              dwTargetID;
  DWORD
                              struTarketZone;
  NET EHOME ZONE
  NET EHOME ZONE
                              struFacePicZone:
  NET EHOME HUMAN FEATURE
                              struHumanFeature:
  DWORD
                              dwStayDuration;
  DWORD
                              dwFacePicLen;
                              byFacePicUrl[MAX URL LEN/*512*/];
  BYTE
                              dwBackgroudPicLen;
  DWORD
  BYTE
                              byBackgroudPicUrl[MAX_URL_LEN/*512*/];
  BYTE
                              byRetransFlag;
```

# ISUPSDK(通用) 开发指南

BYTE byTimeDiffH;
BYTE byTimeDiffM;
BYTE byRes[61];

} NET EHOME FACESNAP REPORT, \*LPNET EHOME FACESNAP REPORT;

# 成员

#### dwSize

结构体大小。

## byDeviceID

设备注册 ID。

### dwVideoChannel

视频通道号。

### byAlarmTime

报警时间,格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

### dwFacePicID

人脸图片 ID。

### dwFaceScore

人脸评分,范围为从1到100。

### dwTargetID

目标 ID。

#### struTarketZone

目标区域的归一化坐标,数值为当前画面尺寸百分比  $\times$  1000,精确到小数点后三位。可参见结构体  $NET\_EHOME\_ZONE$ 。

## struFacePicZone

人脸子图区域的归一化坐标,数值为当前画面尺寸百分比  $\times$  1000,精确到小数点后三位。可参见结构体  $NET\_EHOME\_ZONE$  。

#### struHumanFeature

人体属性,详参见 NET\_EHOME\_HUMAN\_FEATURE 。

### dwStayDuration

画面停留时间,单位: ms。

### dwFacePicLen

人脸图片大小,单位:字节。

### byFacePicUrl

人脸编号。

## dwBackgroudPicLen

背景图片大小,单位:字节。

## byBackgroudPicUrl

背景图片 URL。

## byRetransFlag

重传标记: 0-实时包, 1-重传包。

### byTimeDiffH

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

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**

保留。

## 15. 1. 54 NET\_EHOME\_FLOW\_COND

流量查询条件结构体

# 结构体定义

# 成员

### bySearchMode

查询模式: 1-按年查询, 2-按月查询, 3-按天查询。

#### **bvRes**

保留,设为0。

## struStartTime

开始时间,参见结构体 NET\_EHOME\_TIME 。

## struStopTime

结束时间,参见结构体 NET\_EHOME\_TIME 。

#### dwStartIndex

查询起始位置,从0开始。

### dwMaxFileCountPer

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

时间类型: 0-设备本地时间, 即设备 OSD 时间; 1-UTC 时间。

### byRes1

保留,设为0。

# 15. 1. 55 NET\_EHOME\_FLOW\_INFO

流量信息结构体

# 结构体定义

```
struct{
  DWORD  dwSize;
  DWORD  dwFlowValue;
  DWORD  dwFlowIndex;
  BYTE  byRes[128];
}NET_EHOME_FLOW_INFO, *LPNET_EHOME_FLOW_INFO;
```

## 成员

## dwSize

结构体大小。

### dwFlowValue

流量值,单位: KB。

## dwFlowIndex

流量索引号,从0开始。

# byRes

保留,设为0。

### 15. 1. 56 NET\_EHOME\_GPS\_CFG

GPS 参数结构体

```
struct {
    DWORD          dwSize;
    DWORD          dwTransInterval;
    DWORD          dwMaxSpeed;
    DWORD          dwMinSpeed;
    BYTE          bEnable;
    BYTE          byRes[63];
} NET_EHOME_GPS_CFG, *LPNET_EHOME_GPS_CFG;
```

## 成员

### dwSize

结构体大小。

### dwTransInterval

GPS 信息上传时间间隔,单位:秒。

## dwMaxSpeed

最大速度,单位: cm/h。

### dwMinSpeed

最小速度,单位: cm/h。

#### bEnable

是否支持: 1-否, 2-是。

#### **b**vRes

保留。

# 15. 1. 57 NET\_EHOME\_GPS\_INFO

GPS 报警信息结构体

### 结构体定义

```
struct{
 DWORD dwSize;
        bySampleTime[MAX DEVICE ID LEN/*32*/];
 char
 char
        byDeviceID[MAX_DEVICE_ID_LEN/*256*/];
        byDivision[2];
 char
        bySatelites;
 BYTE
 BYTE
        byPrecision;
 DWORD dwLongitude;
 DWORD dwLatitude;
 DWORD dwDirection;
 DWORD dwSpeed;
```

```
DWORD dwHeight;
BYTE byRetransFlag;
BYTE byLocateMode;
BYTE byRes1[2];
DWORD dwMileage;
BYTE byRes[56];
}NET_EHOME_GPS_INFO, *LPNET_EHOME_GPS_INFO;
```

## 成员

### dwSize

结构体大小

### bySampleTime

GPS 采样时间, 格式: YYYY-MM-DD HH:MM:SS

### byDeviceID

设备注册 ID。

### byDivision

division[0]值: "E"(东经), "W"(西经); division[1]值: "N"(北纬), "S"(南纬)。

### **bySatelites**

卫星数量。

### byPrecision

精度,其值等于(原始值 × 100)。

#### dwLongitude

经度,其值范围为 0 到 (180 × 3600 × 100),转换公式为 (实际度数 × 3600 × 100 + 实际分数 × 60 × 100 + 实际秒数 × 100)。

#### dwLatitude

纬度, 其值范围为 0 到  $(180 \times 3600 \times 100)$ , 转换公式为 ( 实际度数  $\times 3600 \times 100)$  + 实际分数  $\times 60 \times 100$  + 实际秒数  $\times 100$  )。

### dwDirection

方向,其值范围为 0 (表示正北方向) 到 (359.9  $\times$  100),转换公式为 (实际方向值  $\times$  100)。

## dwSpeed

速度, 其值范围为 0 到 (999.9 × 100000), 转换公式为 (实际速度 × 100000), 单位: cm/h。

#### dwHeight

高度,单位:cm。

### byRetransFlag

重传标志: 0-实时包, 1-重传包。

### byLocateMode

定位模式: 0-自定义定位,1-差分,2-评估,3-无效数据。该参数对版本 3.00 的 NMEA0183 可用。

## byRes1

保留。

### dwMileage

设备里程统计,单位: m。统计结果为整数值,且结果会在每天 00:00:00 或每天设备首次启动时被清零。

## byRes

保留。

## 参见

NET\_EHOME\_ALARM\_MSG

## 15. 1. 58 NET EHOME HD STATUS CHANGED

硬盘状态改变信息结构体

# 结构体定义

```
struct{
  DWORD     dwVolume;
  WORD     wHDNo;
  BYTE     byHDStatus;
  BYTE     byRes[5];
} NET_EHOME_HD_STATUS_CHANGED, *LPNET_EHOME_HD_STATUS_CHANGED;
```

## 成员

#### dwVolume

硬盘容量,单位: MB。

### wHDNo

硬盘号。

### **byHDStatus**

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

保留。

# 15. 1. 59 NET\_EHOME\_HD\_TIMING\_STATUS\_SINGLE

硬盘状态定时上传参数结构体

# 结构体定义

```
struct{
  DWORD    dwHDFreeSpace;
  WORD    wHDNo;
  BYTE    byRes[6];
}NET_EHOME_HD_TIMING_STATUS_SINGLE, *LPNET_EHOME_HD_TIMING_STATUS_SINGLE;
```

## 成员

## dwHDFreeSpace

硬盘可用容量,单位: MB。

#### wHDNo

硬盘号。

### **byRes**

保留。

# 15. 1. 60 NET\_EHOME\_HEATMAP\_REPORT

热度图报警信息结构体

## 结构体定义

```
struct {
 DWORD
                                 dwSize;
                                 byDeviceID[MAX DEVICE ID LEN/*256*/];
  char
                                 dwVideoChannel;
  DWORD
  char
                                 byStartTime[MAX TIME LEN/*32*/];
                                 byStopTime[MAX TIME LEN/*32*/];
  char
                                 struHeatmapValue;
  NET EHOME HEATMAP VALUE
  NET EHOME PIXEL ARRAY SIZE
                                 struPixelArraySize;
                                 byPixelArrayData[MAX_URL_LEN/*512*/];
  BYTE
  BYTE
                                 byRetransFlag;
  BYTE
                                 byTimeDiffH;
 BYTE
                                 byTimeDiffM;
                                 byRes[61];
 BYTE
} NET_EHOME_HEATMAP_REPORT, *LPNET_EHOME_HEATMAP_REPORT;
```

## 成员

### dwSize

结构体大小。

## byDeviceID

设备注册 ID。

### dwVideoChannel

视频通道号。

### byStartTime

开始时间,格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

## byStopTime

结束时间,格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

### struHeatmapValue

热度值,参见结构体 NET\_EHOME\_HEATMAP\_VALUE 。

### struPixelArraySize

热度图数据大小,参见结构体 NET\_EHOME\_PIXEL\_ARRAY\_SIZE 。

## byPixelArrayData

热度图数据编号。

### byRetransFlag

重传标记: 0-实时包, 1-重传包。

### byTimeDiffH

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

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**

保留。

## 15. 1. 61 NET\_EHOME\_HEATMAP\_VALUE

热度值信息结构体

# 结构体定义

struct{

DWORD dwMaxHeatMapValue;

```
DWORD dwMinHeatMapValue;
DWORD dwTimeHeatMapValue;
}NET_EHOME_HEATMAP_VALUE, *LPNET_EHOME_HEATMAP_VALUE;
```

# 成员

## dwMaxHeatMapValue

最大热度值。

## dwMinHeatMapValue

最小热度值。

## dwTimeHeatMapValue

平均热度值。

## 参见

NET\_EHOME\_HEATMAP\_REPORT

## 15. 1. 62 NET\_EHOME\_HUMAN\_FEATURE

人体属性信息结构体

# 结构体定义

```
struct{
BYTE byAgeGroup;
BYTE bySex;
BYTE byEyeGlass;
BYTE byRes[13];
}NET_EHOME_HUMAN_FEATURE, *LPNET_EHOME_HUMAN_FEATURE;
```

## 成员

### byAgeGroup

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

性别: 1-男, 2-女。

### byEyeGlass

是否戴眼镜: 1-否, 2-是。

### **b**vRes

保留。

# 15. 1. 63 NET\_EHOME\_IDENTIFICATION

终端 ID 结构体

## 结构体定义

```
struct{
  BYTE bySerialNumber[32];
  BYTE byIdentifyCode[32];
}NET_EHOME_IDENTIFICATION, *LPNET_EHOME_IDENTIFICATION;
```

# 成员

bySerialNumber

序列号,目前暂不支持。

byIdentifyCode

终端 ID。

# 15. 1. 64 NET\_EHOME\_IMAGE\_CFG

图片质量参数结构体

## 结构体定义

```
struct{
  DWORD  dwSize;
BYTE  byHue;
BYTE  byContrast;
BYTE  byBright;
BYTE  bySaturation;
BYTE  byRes[24];
}NET_EHOME_IMAGE_CFG, *LPNET_EHOME_IMAGE_CFG;
```

## 成员

dwSize

结构体大小。

byHue

色调,取值范围: [0,255]。

byContrast

对比度,取值范围: [0,255]。

byBright

亮度,取值范围:[0,255]。

## bySaturation

饱和度,取值范围: [0,255]。

### **byRes**

保留,设为0。

# 15. 1. 65 NET\_EHOME\_INSERT\_CHARACTER

关于插播文字信息的结构体

## 结构体定义

## 成员

### dwMessageCnt

插播消息内容。

#### struTextMessage

插播的文字消息,详见结构体 NET\_EHOME\_TEXT\_MESSAGE 。

### szMsgPos

文字消息的显示位置:顶部、中间、底部、自定义。

### struPosition

文字消息的自定义位置,详见结构体 *NET\_EHOME\_POSITION\_INFO* 。该成员只在 szMsgPos的值为自定义时有效。

### struCharEffect

字符效果,详见结构体 NET\_EHOME\_CHAR\_EFFECT。

## 15. 1. 66 NET\_EHOME\_INSERT\_EFFECT

插播效果结构体

```
struct{
  NET_EHOME_CHAR_EFFECT struCharEffect;
  DWORD dwPageTime;
  DWORD dwScrollSpeedWeb;
}NET_EHOME_INSERT_EFFECT, *LPNET_EHOME_INSERT_EFFECT;
```

## 成员

struCharEffect

字符效果,参见结构体 NET\_EHOME\_CHAR\_EFFECT。

dwPageTime

页面显示时间。

dwScrollSpeedWeb

画面滚动速度。

## 参见

NET\_EHOME\_INSERT\_INFO\_V20

# 15. 1. 67 NET\_EHOME\_INSERT\_INFO

插播参数结构体

# 结构体定义

## 成员

dwInsertType

插播类型,参见枚举 NET\_EHOME\_PROGRAM\_INSERT\_TYPE 。

struMaterialInfo

素材信息,参见结构体 NET\_EHOME\_INSERT\_TEXT\_INFO。

struProgramInfo

节目信息,参见结构体 NET\_EHOME\_INSERT\_TEXT\_INFO。

### struPosInfo

位置信息,参见结构体 NET\_EHOME\_POSITION\_INFO。

## 15. 1. 68 NET EHOME INSERT INFO V20

插播参数结构体(V20)

# 结构体定义

```
struct {
  DWORD
                                dwInsertType;
  NET EHOME INSERT TEXT INFO
                                struMaterialInfo;
                                struProgramInfo:
  NET EHOME INSERT TEXT INFO
  NET EHOME POSITION INFO
                                struPosInfo;
                                struInsertEffect;
  NET EHOME INSERT EFFECT
  DWORD
                                dwPosMode:
  BYTE
                                byRes[64];
} NET_EHOME_INSERT_INFO_V20, *LPNET_EHOME_INSERT_INFO V20;
```

### 成员

### dwInsertType

插播类型,参见枚举 NET EHOME PROGRAM INSERT TYPE。

#### struMaterialInfo

素材信息,参见结构体 NET\_EHOME\_INSERT\_TEXT\_INFO。

### struProgramInfo

节目信息,参见结构体 NET\_EHOME\_INSERT\_TEXT\_INFO 。

### struPosInfo

位置信息,参见结构体 NET\_EHOME\_POSITION\_INFO。

### struInsertEffect

插播效果,参见结构体 NET\_EHOME\_INSERT\_EFFECT。

#### dwPosMode

坐标类型,参见枚举 NET\_EHOME\_POS\_MODE。

## 15. 1. 69 NET\_EHOME\_INSERT\_TEXT\_INFO

插播素材或节目信息结构体

```
struct {
   DWORD     dwTextNo;
   BYTE     szPlayMode[32];
   DWORD     dwCountNum;
   DWORD     dwPlayDuration;
   BYTE     playEndTime[32];
   DWORD     dwTextSeq;
}NET_EHOME_INSERT_TEXT_INFO, *LPNET_EHOME_INSERT_TEXT_INFO;
```

## 成员

### dwTextNo

插播文件序号,即素材 ID 或节目 ID。

### szPlayMode

播放模式: byTime (按时间插播)或 byEndTime (按结束时间插播)。

### dwCountNum

播放次数, 暂不支持。

### dwPlayDuration

播放时间。

### playEndTime

按照结束时间进行插播。

### dwTextSeq

用于判断文件 ID 与文件是否相互对应。

# 15. 1. 70 NET\_EHOME\_IPADDR

IP 地址结构体

## 结构体定义

```
struct{
  char sIpV4[16];
  char sIpV6[128];
}NET_EHOME_IPADDR, *LPNET_EHOME_IPADDR;
```

## 成员

### sIpV4

设备 IPv4 地址。

## sIpV6

设备 IPv6 地址。

# 15. 1. 71 NET\_EHOME\_IPADDRESS

地址信息结构体

# 结构体定义

```
struct{
  char szIP[128];
  WORD wPort;
  BYTE byRes[2];
}NET_EHOME_IPADDRESS, *LPNET_EHOME_IPADDRESS;
```

## 成员

### szIP

IP 地址。

### wPort

端口号。

### **byRes**

保留,设为0。

# 15. 1. 72 NET\_EHOME\_IPC\_INFO

网络摄像机信息结构体

## 结构体定义

```
struct{
 NET_EHOME_IPC_TYPE
                                 enumIPCType;
                                 dwId;
 DWORD
                                 chanNum;
 DWORD
 NET EHOME ADDRESS TYPE
                                 enumAddressType;
  BYTE
                                 szIpcState[32];
                                 szHostName[32];
 BYTE
                                 szIpVersion[16];
 BYTE
  BYTE
                                 szIpAddress[32];
 BYTE
                                 szIpv6Address[32];
                                 dwPortNo;
  DWORD
                                 dwIpcChanNum;
  DWORD
 BYTE
                                 szUserName[52];
                                 szPassWord[32];
 BYTE
 NET EHOME TRANSMINT PROTOCAL enumTransmitProtocol;
```

## 成员

### enumIPCType

网络摄像机类型,参见 NET\_EHOME\_IPC\_TYPE。

#### dwId

网络摄像机 ID。

#### chanNum

关联节目的通道号。

## enumAddressType

地址类型,参见枚举 NET\_EHOME\_ADDRESS\_TYPE 。

### szIpcState

网络摄像机状态: online (在线), offlibe (离线)。

#### szHostName

网络摄像机的域名。

### szIpVersion

IP 地址版本: v4, v6。

### szIpAddress

IPv4 地址。

## szIpv6Address

IPv6 地址。

### **dwPortNo**

网络摄像机的端口号。

### dwIpcChanNum

网络摄像机的通道号。

#### szUserName

网络摄像机的用户名。

### szPassWord

网络摄像机的密码。

### enumTransmitProtocol

传输协议类型,参见枚举 NET\_EHOME\_TRANSMINT\_PROTOCAL 。

### enumStreamType

码流类型,参见枚举 NET\_EHOME\_STREAM\_TYPE 。

# 15. 1. 73 NET\_EHOME\_LINKAGE\_ALARMOUT

报警输出联动参数结构体

## 结构体定义

```
struct{
  DWORD     dwAnalogAlarmOutNum;
  BYTE     byAnalogAlarmOut[MAX_ANALOG_ALARMOUT/*32*/];
  BYTE     byRes[5000];
} NET_EHOME_LINKAGE_ALARMOUT, *LPNET_EHOME_LINKAGE_ALARMOUT;
```

### 成员

## dwAnalogAlarmOutNum

模拟报警输出数量,只读。

### byAnalogAlarmOut

模拟报警输出状态,用数组下标表示: 0-未启用,1-启用。

### **bvRes**

保留,设为0。

# 15. 1. 74 NET\_EHOME\_LINKAGE\_PTZ

PTZ 联动参数结构体

## 结构体定义

```
struct{
  BYTE
          byUsePreset;
  BYTE
          byUseCurise;
          byUseTrack;
  BYTE
  BYTE
          byRes1;
  WORD
          wPresetNo;
  WORD
          wCuriseNo;
  WORD
          wTrackNo;
          byRes2[6];
}NET_EHOME_LINKAGE_PTZ, *LPNET_EHOME_LINKAGE_PTZ;
```

## 成员

### byUsePreset

是否调用预置点: 0-否, 1-是。

### byUseCurise

是否调用巡航:0-否,1-是。

## byUseTrack

是否调用轨迹: 0-否, 1-是。

### byRes1

保留。

### wPresetNo

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

巡航编号,取值范围从1到16。

### wTrackNo

轨迹编号,取值范围从1到16。

### bvRes2

保留。

# 15. 1. 75 NET\_EHOME\_LISTEN\_PREVIEW\_CFG

预览监听参数结构体

## 结构体定义

## 成员

#### struIPAdress

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 NET\_EHOME\_IPADDRESS。

#### fnNewLinkCB

回调函数,用于接收预览请求的响应报文,参见 PREVIEW\_NEWLINK\_CB。

### pUser

用户数据。

## byLinkMode

监听请求的接入方式: 0-TCP, 1-UDP。

### **byRes**

保留,设为0.

## 相关接口

NET ESTREAM StartListenPreview

# 15. 1. 76 NET\_EHOME\_LISTEN\_VOICETALK\_CFG

语音对讲监听参数结构体

## 结构体定义

## 成员

#### struIPAdress

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 NET\_EHOME\_IPADDRESS。

#### fnNewLinkCB

语音对讲请求的回调函数,参见 VOICETALK\_NEWLINK\_CB。

## pUserData

用户参数。

### byRes

保留,设为0。

# 15. 1. 77 NET\_EHOME\_LOCAL\_ACCESS\_SECURITY

设备接入服务器的安全参数结构体

```
struct {
  DWORD    dwSize;
  BYTE    byAccessSecurity;
  BYTE    byRes[127];
} NET_EHOME_LOCAL_ACCESS_SECURITY, *LPNET_EHOME_LOCAL_ACCESS_SECURITY;
```

## 成员

### dwSize

结构体大小。

### byAccessSecurity

接入方式: 0-兼容模式 (所有版本的 ISUP 都支持), 1-普通模式 (只支持 4.0 以下版本的 ISUP), 2-安全模式 (只支持 4.0 及以上版本的 ISUP)。

### **byRes**

保留,设为0。

## 15. 1. 78 NET\_EHOME\_LOCAL\_GENERAL\_CFG

普通参数结构体

# 结构体定义

```
struct{
   BYTE     byAlarmPictureSeparate;
   BYTE     byRes[127];
} NET_EHOME_LOCAL_GENERAL_CFG, *LPNET_EHOME_LOCAL_GENERAL_CFG;
```

## 成员

### byAlarmPictureSeparate

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**

保留,设为0。

# 15. 1. 79 NET\_EHOME\_LONG\_CFG\_INPUT

长连接参数结构体

## 成员

### fnDataCallBack

长连接回调函数,参见 LongConfigCallBack 。

pUser

用户指针。

**byRes** 

保留。

# 15. 1. 80 NET\_EHOME\_LONG\_CFG\_SEND

通过长连接发送的数据结构体

# 结构体定义

```
struct{
  void *pDataBuffer;
  DWORD dwDataLen;
  BYTE byRes[32];
}NET_EHOME_LONG_CFG_SEND, *LPNET_EHOME_LONG_CFG_SEND;
```

## 成员

## pDataBuffer

保存待发送数据的缓冲区。

dwDataLen

待发送数据的缓冲区大小。

byRes

保留。

# 15. 1. 81 NET\_EHOME\_MAKE\_I\_FRAME

强制 I 帧参数结构体

```
struct{
int iChannel;
DWORD dwStreamType;
BYTE byRes[40];
}NET_EHOME_MAKE_I_FRAME, *LPNET_EHOME_MAKE_I_FRAME;
```

## 成员

### *iChannel*

通道号。

## dwStreamType

码流类型: 0-主码流, 1-子码流, 2-保留。

### **byRes**

保留。

# 15. 1. 82 NET EHOME MANUAL IOOUT CTRL

报警输出手动控制参数结构体

# 结构体定义

```
struct {
  DWORD    dwSize;
  DWORD    dwChan;
  DWORD    dwDelayTime;
  BYTE    byAction;
  BYTE    byRes[19];
}NET_EHOME_MANUAL_IOOUT_CTRL,*LPNET_EHOME_MANUAL_IOOUT_CTRL;
```

# 成员

### dwSize

结构体大小。

### dwChan

报警输出号,从0开始。

### dwDelayTime

报警输出持续时间(s),值为0表示一直输出。

### byAction

控制类型: 0-关闭报警输出, 1-开启报警输出。

### byRes

保留,设为0。

## 15. 1. 83 NET\_EHOME\_MPDATA

车载设备客流统计数据结构体

# 结构体定义

```
struct{
   BYTE byIndex;
BYTE byVideoChannel;
BYTE byRes;
BYTE byLevel;
char byStarttime[MAX_TIME_LEN/*32*/];
char byStoptime[MAX_TIME_LEN/*32*/];
DWORD dwEnterNum;
DWORD dwLeaveNum;
DWORD dwCount;
}NET_EHOME_MPDATA, *LPNET_EHOME_MPDATA;
```

### 成员

## byIndex

车门编号,从1开始。

### byVideoChannel

视频通道号,从1开始。

#### **byRes**

保留。

### byLeve1

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)

#### bvStarttime

开始统计时间/开门时间,格式:YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

## byStoptime

停止统计时间/关门时间,格式: YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

### dwEnterNum

统计时间段内进入人数。

#### dwLeaveNum

统计时间内离开人数。

### dwCount

当前时刻车内人数。

# 15. 1. 84 NET EHOME MPGPS

车载客流统计的 GPS 定位信息结构体

# 结构体定义

```
struct{
  LONG iLongitude;
  LONG iLatitude;
  LONG iSpeed;
  LONG iDirection;
} NET_EHOME_MPGPS, *LPNET_EHOME_MPGPS;
```

## 成员

iLongitude

经度信息。

iLatitude

维度信息。

**iSpeed** 

速度。

iDirection

方向。

# 15. 1. 85 NET\_EHOME\_NETWORK\_CFG

网络参数结构体

## 结构体定义

```
struct {dwSize;DWORDdwSize;NET_EHOME_ETHERNETstruEtherNet;NET_EHOME_IPADDRstruGateWayIP;NET_EHOME_IPADDRstruMultiCastIP;NET_EHOME_IPADDRstruDDNSServer1IP;NET_EHOME_IPADDRstruDDNSServer2IP;NET_EHOME_IPADDRstruAlarmHostIP;
```

WORD wAlarmHostPort;
WORD wIPResolverPort;
NET\_EHOME\_IPADDR struIPResolver;
NET\_EHOME\_PPPOECFG struPPPoE;
WORD wHTTPPort;
BYTE byRes[674];
}NET\_EHOME\_NETWORK\_CFG, \*LPNET\_EHOME\_NETWORK\_CFG;

## 成员

#### dwSize

结构体大小。

#### struEtherNet

以太网口信息,参见结构体 NET EHOME ETHERNET。

### struGateWayIP

网关地址,参见结构体 NET EHOME IPADDR。

## struMultiCastIP

多播地址,参见结构体 NET\_EHOME\_IPADDR。

#### struDDNSServer1IP

DDNS1的 IP 地址, 参见结构体 NET EHOME IPADDR。

### struDDNSServer2IP

DDNS2的 IP 地址,参见结构体 NET EHOME IPADDR。

### struAlarmHostIP

报警主机 IP 地址,参见结构体 NET EHOME IPADDR。

### wAlarmHostPort

报警主机端口号。

## wIPResolverPort

解析服务器端口号。

#### struIPResolver

解析服务器地址,参见结构体 NET\_EHOME\_IPADDR。

#### struPPPoE

PPPoE 参数,参见结构体 NET\_EHOME\_PPPOECFG。

#### wHTTPPort

HTTP 通信端口号。

### byRes

保留。

# 15. 1. 86 NET\_EHOME\_NEWLINK\_CB\_MSG

预览请求参数结构体

## 结构体定义

# 成员

szDeviceID

设备 ID。

iSessionID

设备取流会话 ID。

dwChannelNo

通道号。

byStreamType Stream

码流类型: 0-主码流, 1-子码流。

sDeviceSerial

设备序列号。

**byRes** 

保留,设为0。

# 15. 1. 87 NET\_EHOME\_NOTICE\_PICURL

图片 URL 信息结构体

# 结构体定义

```
struct{
  DWORD    dwSize;
  char    byDeviceID[MAX DEVICE ID LEN/*256*/];
```

```
WORD
           wPicType;
           wAlarmType;
  WORD
  DWORD
           dwAlarmChan:
           byAlarmTime[MAX TIME LEN/*32*/];
  char
           dwCaptureChan;
  DWORD
  char
           byPicTime[MAX TIME LEN/*32*/];
           byPicUrl[MAX URL LEN/*512*/];
  char
           dwManualSnapSeq;
  DWORD
           byRetransFlag;
  BYTE
  BYTE
           byTimeDiffH;
           byTimeDiffM;
  BYTE
  BYTE
           byRes[29];
NET EHOME NOTICE PICURL, *LPNET EHOME NOTICE PICURL;
```

## 成员

#### dwSize

结构体大小。

### byDeviceID

设备注册 ID。

### wPicType

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

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

报警通道号,只有当wPicType设为1时,该参数有效。

### byAlarmTime

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。

### dwCaptureChan

抓拍通道号。

### byPicTime

图片显示时间,格式:YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

### byPicUrl

图片 URL。

#### dwManualSnapSeq

请求序列号,只有当PicType设为2或3时,该参数有效。

## **byRetransFlag**

重传标记: 0-实时包, 1-重传包。

## byTimeDiffH

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

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**

保留。

## 参见

NET EHOME ALARM MSG

# 15. 1. 88 NET\_EHOME\_NOTIFY\_FAIL\_INFO

异步失败通知信息结构体

# 结构体定义

```
struct{
 DWORD
           dwSize;
           byDeviceID[MAX DEVICE ID LEN[/*256*/];
  char
           wFailedCommand;
  WORD
  WORD
           wPicType;
           dwManualSnapSeq;
  DWORD
  BYTE
           byRetransFlag:
  BYTE
           byRes[31];
}NET EHOME NOTIFY FAIL INFO, *LPNET EHOME NOTIFY FAIL INFO;
```

## 成员

#### dwSize

结构体大小。

#### byDeviceID

设备注册 ID。

## wFailedCommand

失败的命令: 1-抓图失败。

#### wPicType

图片类型: 0-计划抓图且上传中心,1-报警触发抓图且上传中心,2-手动抓图且上传中心,3-下载的图片,4-外部设备抓图且上传中心。该参数只有当 wFailedCommand 设为1时有效。

## dwManualSnapSeq

请求序列号,该参数只有当wPicType为2或3时有效。

#### byRetransFlag

重传标志: 0-实时包, 1-重传包。

## byRes

保留。

## 参见

NET\_EHOME\_ALARM\_MSG

# 15. 1. 89 NET EHOME PASSTHROUGH PARAM

协议参数结构体

# 结构体定义

## 成员

#### dwSequence

报文序号。

#### dwUUID

会话 ID。

#### **byRes**

保留。

# 15. 1. 90 NET\_EHOME\_PASSWORD\_CFG

终端密码参数结构体

## 结构体定义

```
struct{
  BYTE     sz01dPwd[32];
  BYTE     szNewPwd[32];
}NET_EHOME_PASSWORD_CFG, *LPNET_EHOME_PASSWORD_CFG;
```

## 成员

sz01dPwd

旧密码。

szNewPwd

新密码。

# 15. 1. 91 NET\_EHOME\_PIC\_CFG

OSD 显示参数结构体

## 结构体定义

```
struct{
 DWORD
           dwSize:
 BYTE
           byChannelName[NAME LEN/*32*/];
 BYTE
           bIsShowChanName;
  WORD
           wChanNameXPos:
  WORD
           wChanNameYPos;
 BYTE
           bIsShowOSD;
           wOSDXPos;
  WORD
  WORD
           wOSDYPos:
 BYTE
           byOSDType;
           byOSDAtrib;
 BYTE
 BYTE
           byRes1[2];
 BYTE
           bIsShowWeek;
 BYTE
           byRes2[64];
}NET EHOME PIC CFG, *LPNET EHOME PIC CFG;
```

# 成员

dwSize

结构体大小。

byChannelName

通道名称,长度不超过31字节。

bIsShowChanName

是否显示通道名称: 0-否, 1-是。

#### wChanNameXPos

通道名称显示的 X 轴坐标,按照  $704 \times 576$  来配置,坐标值为 16 的倍数。

#### wChanNameYPos

通道名称显示的 Y 轴坐标,按照  $704 \times 576$  来配置,坐标值为 16 的倍数。

#### bIsShowOSD

是否显示日期信息: 0-否, 1-是。

#### wOSDXPos

日期信息显示的 X 轴坐标,按照  $704 \times 576$  来配置,坐标值为 16 的倍数。

#### **w**OSDYPos

日期信息显示的 Y 轴坐标,按照 704 × 576 来配置,坐标值为 16 的倍数。

## by0SDType

日期显示格式: 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

## byRes1

保留,设为0。

#### bIsShowWeek

是否显示星期: 0-否, 1-是。

#### **byRes**

保留,设为0。

## 15. 1. 92 NET\_EHOME\_PIC\_FILE

图片文件查找结果结构体

```
struct{
 DWORD
                    dwSize:
  char
                    sFileName[MAX FILE NAME LEN/*100*/];
  NET EHOME TIME
                    struPicTime;
                    dwFileSize:
  DWORD
  DWORD
                    dwFileMainType;
  DWORD
                    dwFileIndex:
  BYTE
                    byTimeDiffH;
  BYTE
                    byTimeDiffM:
                    byRes[126];
  BYTE
}NET EHOME PIC FILE, *LPNET EHOME PIC FILE;
```

#### dwSize

结构体大小。

## szFileName

图片文件名。

#### struPicTime

图片生成时间,参见结构体 NET EHOME TIME 。

#### dwFileSize

图片文件大小。

## dwFileMainType

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) -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

图片文件编号,从0开始。

#### bvTimeDiffH

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

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。

#### **bvRes**

保留,设为0。

# 15. 1. 93 NET\_EHOME\_PIC\_FILE\_COND

图片文件查找条件结构体

## 结构体定义

```
struct{
  DWORD
                     dwChanne1:
  DWORD
                     dwPicTvpe:
                    struStartTime;
  NET EHOME TIME
  NET EHOME TIME
                     struStopTime;
  DWORD
                     dwStartIndex:
  DWORD
                     dwMaxFileCountPer;
  BYTE
                     byLocalOrUTC;
  BYTE
                     byRes[63];
}NET EHOME PIC FILE COND, *LPNET EHOME PIC FILE COND;
```

## 成员

#### dwChannel

通道号。

## dwPicType

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) -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

开始时间,参见结构体 NET\_EHOME\_TIME。

#### struStopTime

结束时间,参见结构体 NET\_EHOME\_TIME 。

#### dwStartIndex

查询起始位置,从0开始。

#### dwMaxFileCountPer

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

时间类型: 0-设备本地时间,即设备 OSD 时间; 1-UTC 时间。

#### byRes

保留,设为0。

# 15. 1. 94 NET\_EHOME\_PIXEL\_ARRAY\_SIZE

热度图大小信息结构体

# 结构体定义

## 成员

dwLineValue

像素点行值。

dwColumnValue

像素点列值。

# 15. 1. 95 NET EHOME PLANCTRLPARAM

定时控制计划参数结构体

# 结构体定义

```
struct{
  NET_EHOME_PLANCTRL_TYPE enumPlanType;
  BOOL bEnable;
} NET_EHOME_PLAN_CTRLPARAM, *LPNET_EHOME_PLAY_CTRLPARAM;
```

## 成员

enumPlanType

控制类型,参见枚举 NET\_EHOME\_PLANCTRL\_TYPE 。

bEnable

是否启用控制计划。

## 参见

NET\_EHOME\_TERMINAL\_CONTROL\_V20

# 15. 1. 96 NET\_EHOME\_PLAYBACK\_DATA\_CB\_INFO

回放数据的回调信息结构体

## 结构体定义

```
struct {
  BYTE     dwType;
  void *pData;
  DWORD     dwDataLen;
  BYTE     byRes[128];
} NET_EHOME PLAYBACK_DATA_CB_INFO, *LPNET_EHOME_PLAYBACK_DATA_CB_INFO;
```

## 成员

# byDataType Data

数据类型: 0-码流头部, 1-码流数据。

## pData Data

保存码流头部或码流数据所需的缓冲区。

#### dwDataLen

码流头部或码流数据缓冲区大小。

#### **byRes**

保留。

# 15. 1. 97 NET\_EHOME\_PLAYBACK\_DATA\_CB\_PARAM

回放数据的回调参数结构体

# 结构体定义

# 成员

## fnPlayBackDataCB

回放数据的回调函数,参见 PLAYBACK\_DATA\_CB。

#### pUserData

用户参数。

## byStreamFormat

封装格式: 0-PS

byRes

保留,设为0。

# 15. 1. 98 NET\_EHOME\_PLAYBACK\_INFO\_IN

回放请求的输入参数结构体

# 结构体定义

```
struct{
 DWORD
                          dwSize;
 DWORD
                          dwChannel;
  BYTE
                          byPlayBackMode;
                          byStreamPackage;
 BYTE
  BYTE
                          byRes[2];
  union {
    BYTE
                         byLen[512];
    struct {
                         szFileName[MAX FILE NAME LEN/*100*/];
      char
      DWORD
                         dwSeekType:
      DWORD
                         dwFileOffset;
      DWORD
                         dwFileSpan;
    } struPlayBackbyName;
    struct {
      NET_EHOME_TIME
                         struStartTime;
      NET EHOME TIME
                         struStopTime;
      BYTE
                         byLocalOrUTC;
      BYTE
                         byDuplicateSegment;
   } struPlayBackbyTime;
  }unionPlayBackMode;
  NET EHOME IPADDRESS
                          struStreamSever;
INET EHOME PLAYBACK INFO IN, *LPNET EHOME PLAYBACK INFO IN;
```

# 成员

#### dwSize

结构体大小。

#### dwChannel

回放通道号。

## byPlayBackMode

回放和下载模式: 0-按文件名, 1-按时间。

## byStreamPackage

码流回放类型: 0-PS (默认), 1-RTP。

## **byRes**

保留。

#### unionPlayBackMode

回放下载模式参数联合体, 详见以下联合体参数描述。

#### byLen

联合体大下,512字节。

#### struPlayBackbyName

按文件名回放的参数结构体,详见以下结构体参数描述。

#### szFileName

讲行回放的文件名。

## dwSeekType

偏移量计算方式: 0-按字节长度, 1-按时间(秒数)。

#### dwFileOffset

文件偏移量,开始回放或下载的位置。

## dwFileSpan

下载的文件大小,为0时,表示下载整个文件。

## struPlayBackbyTime

按时间回放的参数结构体,详见以下结构体参数描述。

#### struStartTime

开始回放的时间,详见结构体 NET\_EHOME\_TIME 。

## struStopTime

停止回放的时间,详见结构体 NET\_EHOME\_TIME 。

## byLocalOrUTC

时间类型: 0-设备本地时间, 即设备 OSD 时间; 1-UTC 时间。

## byDuplicateSegment

重复时间段的位置: 0-重复时间段的前段,1-重复时间段的后段。当 byLocal0rUTC 为 1 时,该参数无效。

#### struStreamSever

流媒体服务器的地址信息,参见结构体 NET\_EHOME\_IPADDRESS。

## 备注

目前仅支持按文件名回放。

# 15. 1. 99 NET\_EHOME\_PLAYBACK\_INFO\_OUT

回放请求的输出参数结构体

## 结构体定义

```
struct{
  LONG iSessionID;
  LONG lHandle;
  BYTE byRes[124];
} NET_EHOME_PLAYBACK_INFO_OUT, *LPNET_EHOME_PLAYBACK_INFO_OUT;
```

## 成员

#### iSessionID

设备返回的回放请求会话 ID: 0-无效。

#### 1Handle

异步回调的消息句柄。

#### **byRes**

保留,设为0。

# 15. 1. 100 NET EHOME PLAYBACK LISTEN PARAM

回放的监听参数结构体

## 结构体定义

## 成员

#### struIPAdress

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 NET\_EHOME\_IPADDRESS 。

#### fnNewLinkCB

回放请求的回调函数,参见 PLAYBACK\_NEWLINK\_CB。

## pUserData

用户参数。

#### byLinkMode

监听请求的接入协议类型: 0-TCP, 1-UDP (保留)

#### **bvRes**

保留,设为0。

## 备注

若设备不支持 4.0 及以上版本的 ISUP,则 **1SessionID** 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。

# 15. 1. 101 NET\_EHOME\_PLAYBACK\_NEWLINK\_CB\_INFO

回放请求的回调信息结构体

# 结构体定义

```
struct{
  char     szDeviceID[NET_EHOME_DEVICEID_LEN/*256*/];
LONG     1SessionID;
DWORD     dwChannelNo;
char     sDeviceSeria1[NET_EHOME_SERIAL_LEN/*12*/];
BYTE     byRes[108];
} NET_EHOME_PLAYBACK_NEWLINK_CB_INFO, *LPNET_EHOME_PLAYBACK_NEWLINK_CB_INFO;
```

# 成员

#### szDeviceID

设备 ID。

#### 1SessionID

回放请求的设备会话 ID,由 NET\_ECMS\_StartPlayBack 返回。若无效,则设为 0。

#### dwChannelNo

设备通道号,0-无效。

#### sDeviceSerial

设备序列号。

## **byRes**

保留,设为0。

# 15. 1. 102 NET\_EHOME\_PLAYBACK\_PAUSE\_RESTART\_PARAM

回放暂停或恢复参数结构体

## 结构体定义

## 成员

#### 1SessionID

回放的会话 ID。

#### 1Handle

输出参数指针,在异步模式中作为异步回调的标识,对应 NET\_EHOME\_CMSCB\_DATA 中的 dwHandle。

#### **bvRes**

保留。

# 15. 1. 103 NET EHOME PLAYER PARAM

终端播放参数结构体

# 结构体定义

```
struct {

BYTE szCfgType[32];

DWORD dwplayerVolume;

NET_EHOME_DISPLAY_PARAM struDisplayParam;

NET_EHOME_PASSWORD_CFG struPasswordCfg;

}NET_EHOME_PLAYER_PARAM, *LPNET_EHOME_PLAYER_PARAM;
```

## 成员

#### szCfgType

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)。

## dwplayerVolume

播放音量。

## struDisplayParam

显示参数,参见结构体 NET EHOME DISPLAY PARAM。

## struPasswordCfg

终端密码,参见结构体 NET\_EHOME\_PASSWORD\_CFG 。

## 15. 1. 104 NET EHOME POINT

坐标参数结构体

# 结构体定义

```
struct {
  DWORD  dwX;
  DWORD  dwY;
  BYTE  byRes[4];
}NET_EHOME_POINT, *LPNET_EHOME_POINT;
```

## 成员

#### dwX

X轴坐标信息。

#### dwY

Y轴坐标信息。

## **byRes**

保留,设为0。

# 15. 1. 105 NET\_EHOME\_POSITION\_INFO

位置信息结构体

```
struct{
   DWORD dwPositionX;
   DWORD dwPositionY;
   DWORD dwHeight;
   DWORD dwWidth;
}NET_EHOME_POSITION_INFO, *LPNET_EHOME_POSITION_INFO;
```

## dwPositionX

位置的X轴坐标。

#### dwPositionY

位置的Y轴坐标。

## dwHeight

高度。

## dwWidth

宽度。

# 15. 1. 106 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

# 结构体定义

```
struct{
 BYTE
          szServerIP[32];
 DWORD
          szServerPort;
 DWORD
          dwTerminalID:
  DWORD
          dwXm1UniqueSeq;
 DWORD
          dwXmlSizeH;
          dwXmlSizeL;
 DWORD
          byStorageId[128];
 BYTE
 BYTE
          byRes[16];
}NET_EHOME_POST_PUBLISH_XML, *LPNET_EHOME_POST_PUBLISH_XML;
```

# 成员

## szServerIP

服务器 IP 地址。

## szServerPort

服务器端口号。

#### dwTerminalID

终端 ID, 随进程返回。

## dwXm1UniqueSeq

XML 文件 ID。

dwXm1SizeH

XML 文件大小, 高 32 位表示。

#### dwXm1SizeL

XML 文件大下, 低 32 位表示。

## byStorageId

XML 文件的保存路径。

## byRes

保留。

# 15. 1. 107 NET\_EHOME\_POST\_SCHEDULE

日程下发参数结构体

# 结构体定义

```
struct{
  BYTE
          byServerIP[32];
  DWORD
          dwServerPort:
          dwScheduleID;
  DWORD
  DWORD
          dwScheduleSeq;
          byIsDefaultSchedual;
  BOOL
  BYTE
          byRes[2];
          dwTimingSchedualID;
  DWORD
  BYTE
          byEffectiveTime[32];
} NET EHOME POST SCHEDULE, *LPNET EHOME POST SCHEDULE;
```

## 成员

## byServerIP

服务器 IP 地址。

#### dwServerPort

服务器端口号。

## dwScheduleID

新建日程是分配的日程 ID。

## dwScheduleSeq

标示日程是否被修改,是否可以继续传输,只针对 ANR 有效。

## byIsDefaultSchedual

标示是否为垫片日程。

## byRes

保留。

## dwTimingSchedualID

日程 ID。

## byEffectiveTime

日程开始播放的时间。

# 15. 1. 108 NET\_EHOME\_PPPOECFG

PPPoE 参数结构体

# 结构体定义

## 成员

#### dwPPPoE

是否启用 PPPoE: 1-是, 0-否。

#### sPPPoEUser

PPPoE 用户名。

#### sPPPoEPassword

PPPoE 密码。

#### struPPPoEIP

PPPoE 的 IP 地址,参见结构体 NET\_EHOME\_IPADDR。

# 15. 1. 109 NET\_EHOME\_PRESET\_PARAM

预置点参数结构体

```
struct{
  DWORD    dwSize;
  BYTE    byPresetCmd;
  BYTE    byRes1[3];
  DWORD    dwPresetIndex;
```

```
BYTE byRes2[32];
}NET_EHOME_PRESET_PARAM, *LPNET_EHOME_PRESET_PARAM;
```

#### dwSize

结构体大小。

#### byPresetCmd

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

保留,设为0。

#### dwPresetIndex

预置点编号。

#### byRes2

保留,设为0。

# 15. 1. 110 NET\_EHOME\_PREVIEW\_CB\_MSG

预览回调数据结构体

# 结构体定义

```
struct{
  BYTE    byDataType;
  BYTE    byRes1[3];
  void *pRecvdata;
  DWORD   dwDataLen;
  BYTE   byRes2[128];
} NET_EHOME_PREVIEW_CB_MSG, *LPNET_EHOME_PREVIEW_CB_MSG;
```

## 成员

## byDataType

数据类型: 1-NET\_DVR\_SYSHEAD(码流头部), 2-NET\_DVR\_STREAMDATA(码流数据)。

## byRes1

保留。

#### pRecydata

保存码流头部或码流数据的缓冲区。

#### dwDataLen

码流头部或码流数据缓冲区大小。

## byRes2

保留。

# 15. 1. 111 NET\_EHOME\_PREVIEW\_DATA\_CB\_PARAM

预览回调参数结构体

# 结构体定义

# 成员

#### fnPreviewDataCB

预览的回调函数,参见 PREVIEW DATA CB。

## pUserData

用户参数。

## byStreamFormat

码流封装格式: 0-PS

#### **b**vRes

保留,设为0。

# 15. 1. 112 NET\_EHOME\_PREVIEWINFO\_IN\_V11

预览请求的输入参数结构体

```
struct{
                         iChannel;
  int
                         dwStreamType;
  DWORD
  DWORD
                         dwLinkMode;
 NET_EHOME_IPADDRESS
                         struStreamSever;
 BYTE
                         byDelayPreview;
 BYTE
                         byEncrypt;
                         byRes[30];
 BYTE
}NET_EHOME_PREVIEWINFO_IN_V11, *LPNET_EHOME_PREVIEWINFO_IN_V11;
```

## *iChannel*

通道号。

## dwStreamType

码流类型: 0-主码流, 1-子码流, 2-第三码流。

#### dwLinkMode

接入方式: O-TCP, 1-UDP, 2-HRUDP (可靠传输)

#### struStreamSever

流媒体服务器的地址信息,参见结构体 NET EHOME IPADDRESS。

#### byDelayPreview

预览模式: 0-获取实时码流, 1-获取延迟码流。

#### byEncrypt

码流是否加密: 0-否, 1-是。

### byRes

保留,设为0。

# 15. 1. 113 NET\_EHOME\_PREVIEWINFO\_OUT

预览请求的输出参数结构体

# 结构体定义

## 成员

#### 1SessionID

预览请求的会话 ID。

#### 1Handle

异步取流回调的消息句柄。

#### **byRes**

保留,设为0。

# 15. 1. 114 NET\_EHOME\_PT\_PARAM

监听代理参数结构体

## 结构体定义

## 成员

#### struIP

监听地址信息(IP地址和端口号),参见结构体 NET\_EHOME\_IPADDRESS。

## byProtocolType

协议类型: 0-TCP

# byProxyType

代理类型: 0-海康私有协议代理, 1-HTTP代理。

#### **byRes**

保留。

## 15. 1. 115 NET EHOME PTXML PARAM

传输参数结构体

```
struct{
  void
              *pRequestUr1;
  DWORD
              dwRequestUrlLen;
              *pCondBuffer:
  void
  DWORD
              dwCondSize:
              *pInBuffer;
  void
              dwInSize;
  DWORD
              *pOutBuffer:
  void
  DWORD
              dwOutSize:
              dwReturnedXMLLen;
  DWORD
              dwRecvTimeOut;
  DWORD
  BYTE
              byRes[32];
}NET_EHOME_PTXML_PARAM, *LPNET_EHOME_PTXML_PARAM;
```

# pRequestUr1

请求 URL。

## dwRequestUrlLen

请求 URL 大小。

## pCondBuffer

保存条件参数(XML或JSON格式)的缓冲区。

#### dwCondSize

条件缓冲区大小。

## pInBuffer

保存输入参数(XML或 JSON 格式)的缓冲区。

#### dwInSize

输入缓冲区大小。

## pOutBuffer

保存输出参数(XML或JSON格式)的缓冲区。

#### dwOutSize

输出缓冲区大小。

## dwReturnedXMLLen

实际接收 XML 或 JSON 数据大小。

## dwRecvTimeOut

接收超时时间。

## byRes

保留,设为0。

## 15. 1. 116 NET EHOME PTZ PARAM

PTZ 参数结构体

struct{	
DWORD	dwSize;
BYTE	byPTZCmd;
BYTE	<pre>byAction;</pre>
BYTE	bySpeed;

# ISUPSDK (通用) 开发指南

BYTE **byRes**[29];

}NET\_EHOME\_PTZ\_PARAM, \*LPNET\_EHOME\_PTZ\_PARAM;

# 成员

# dwSize

结构体大小。

# byPTZCmd

PTZ 控制命令,参见以下表格。

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

# byAction

PTZ 控制: 0-开始, 1-停止。

## bySpeed

PTZ 速度,取值范围从 0 到 70。值越大,代表速度越快。

## byRes

保留。

# 15. 1. 117 NET\_EHOME\_PUBLISH\_SERVERADDR

信息发布服务器的地址信息结构体

# 结构体定义

```
struct{
 NET EHOME ADDRFORMAT CMD enumAddressingFormatType;
                            szIpAddress[NET EHOME MAX TYPE LEN/*32*/];
                            szIpv6Address[NET_EHOME_MAX_TYPE_LEN/*32*/];
 BYTE
                            szHostName[NET_EHOME_MAX_TYPE_LEN/*32*/];
 BYTE
 DWORD
                            dwPortNo;
 BYTE
                            szUserName[NET_EHOME_MAX_TYPE_LEN/*32*/];
 BYTE
                            szPassword[NET_EHOME_MAX_TYPE_LEN/*32*/];
                            bRegStatus;
 BOOL
NET EHOME PUBLISH SERVERADDR, *LPNET EHOME PUBLISH SERVERADDR;
```

## 成员

#### enumAddressingFormatType

地址类型,参见枚举 NET EHOME ADDRFORMAT CMD。

#### szIpAddress

IP 地址。

#### szIpv6Address

IPv6 地址。

#### szHostName

域名。

#### dwPortNo

端口号。

#### szUserName

用户名。

#### szPassword

密码。

## bRegStatus

注册状态。

# 15. 1. 118 NET\_EHOME\_PUSHPLAYBACK\_IN

回放码流传输请求的输入参数结构体

## 结构体定义

```
struct {
  DWORD    dwSize;
  LONG    1SessionID;
  BYTE    byKeyMD5[32];
  BYTE    byRes[96];
}NET_EHOME_PUSHPLAYBACK_IN, *LPNET_EHOME_PUSHPLAYBACK_IN;
```

## 成员

dwSize

结构体大小。

1SessionID

会话 ID, 由 NET\_ECMS\_StartPlayBack 返回。

byKeyMD5

码流加密密钥,通过两次 MD5 计算获得。

**byRes** 

保留,设为0。

# 15. 1. 119 NET\_EHOME\_PUSHPLAYBACK\_OUT

回放码流传输请求的输出参数结构体

# 结构体定义

```
struct {
  DWORD    dwSize;
  LONG    1Handle;
  BYTE    byRes[124];
} NET_EHOME_PUSHPLAYBACK_OUT, *LPNET_EHOME_PUSHPLAYBACK_OUT;
```

## 成员

dwSize

结构体大小。

#### 1Handle

异步回调的消息句柄。

## **byRes**

保留,设为0。

# 15. 1. 120 NET\_EHOME\_PUSHSTREAM\_IN

实时流传输请求的输入参数结构体

# 结构体定义

```
struct{
  DWORD   dwSize;
  LONG   1SessionID;
  BYTE   byRes[128];
} NET_EHOME_PUSHSTREAM_IN, *LPNET_EHOME_PUSHSTREAM_IN;
```

## 成员

## dwSize

结构体大小。

## 1SessionID

预览请求的会话 ID,由 NET\_ECMS\_StartGetRealStreamV11 返回。

## **byRes**

保留,设为0。

# 15. 1. 121 NET\_EHOME\_PUSHSTREAM\_OUT

实时流传输请求的输出参数结构体

# 结构体定义

```
struct{
  DWORD    dwSize;
  LONG    1Handle;
  BYTE    byRes[124];
}NET_EHOME_PUSHSTREAM_OUT, *LPNET_EHOME_PUSHSTREAM_OUT;
```

# 成员

## dwSize

结构体大小。

#### 1Handle

异步取流回调的消息句柄。

## **byRes**

保留,设为0。

# 15. 1. 122 NET\_EHOME\_PUSHVOICE\_IN

音频码流传输请求的输入参数结构体

## 结构体定义

```
struct{
  DWORD    dwSize;
  LONG    1SessionID;
  BYTE    byToken[64];
  BYTE    byRes[64];
} NET_EHOME_PUSHVOICE_IN, *LPNET_EHOME_PUSHVOICE_IN;
```

# 成员

#### dwSize

结构体大小。

#### 1SessionID

语音对讲请求的会话 ID,由 NET\_ECMS\_StartVoiceWithStmServer 返回。

#### byToken

令牌信息。

#### **byRes**

保留,设为0。

# 15. 1. 123 NET\_EHOME\_PUSHV01CE\_OUT

音频码流传输请求的输出参数结构体

```
struct{
  DWORD  dwSize;
  LONG    1Handle;
  BYTE   byRes[124];
}NET_EHOME_PUSHVOICE_OUT, *LPNET_EHOME_PUSHVOICE_OUT;
```

## dwSize

结构体大小。

#### 1Handle

异步音频码流回调的消息句柄。

## **byRes**

保留,设为0。

# 15. 1. 124 NET\_EHOME\_PZIN\_PARAM

变倍参数结构体

# 结构体描述

```
struct{
  DWORD      dwSize;
  BYTE      byAction;
  BYTE      byRes1[3];
  NET_EHOME_ZONE     struArea;
  BYTE      byRes2[32];
} NET_EHOME_PZIN_PARAM, *LPNET_EHOME_PZIN_PARAM;
```

# 成员

#### dwSize

结构体大小。

#### byAction

变倍类型: 0-缩小, 1-放大。

## byRes1

保留,设为0。

#### struArea

框选范围参数,参见结构体 NET\_EHOME\_ZONE 。

#### byRes2

保留,设为0。

# 15. 1. 125 NET\_EHOME\_QUEUE\_DATA

呼叫号码数据结构体

## 结构体定义

```
struct{
  char    szQueueDataValue[32];
  DWORD    dwQueueDataId;
}NET_EHOME_QUEUE_DATA;
```

## 成员

#### szQueueDataValue

呼叫号码。

## dwQueueDataId

呼叫号码 ID。

# 15. 1. 126 NET\_EHOME\_QUEUE\_DATALIST

呼叫号码列表结构体

## 结构体定义

```
struct {
   NET_EHOME_QUEUE_DATA queueData[20];
}NET_EHOME_QUEUE_DATALIST;
```

## 成员

## queueData

呼叫号码,参见结构体 NET\_EHOME\_QUEUE\_DATA。

# 15. 1. 127 NET\_EHOME\_QUEUE\_INFO

叫号管理信息结构体

#### dwMaterialId

素材 ID。

#### dwQueueId

呼叫号码 ID。

## enumDataType

命令类型,参见枚举 NET\_EHOME\_THIRD\_PARTY\_DATA\_TYPE 。

## enumRefreshType

数据刷新类型,参见枚举 NET\_EHOME\_REFRESH\_TYPE 。

#### struItemDataList

呼叫号码列表,参见结构体 NET EHOME QUEUE ITEM DATALIST。

# 15. 1. 128 NET\_EHOME\_QUEUE\_ITEM\_DATALIST

呼叫号码列表结构体

# 结构体定义

## 成员

#### dwQueueDataListCnt

列表的行数。

#### dwQueueDataCnt

列表的列数。

#### struQueueDatalist

呼叫号码列表,参见结构体 NET EHOME QUEUE DATALIST。

## 15. 1. 129 NET\_EHOME\_REC\_FILE

视频文件查找结果结构体

## 结构体定义

```
struct{
  DWORD
                     dwSize:
                     sFileName[MAX FILE NAME LEN/*100*/];
  char
  NET EHOME TIME
                    struStartTime;
                     struStopTime;
  NET EHOME TIME
  DWORD
                     dwFileSize:
  DWORD
                     dwFileMainType;
  DWORD
                     dwFileSubType;
  DWORD
                     dwFileIndex;
                     byTimeDiffH;
  BYTE
  BYTE
                    byTimeDiffM;
  BYTE
                    byRes[126];
} NET EHOME REC FILE, *LPNET EHOME REC FILE;
```

# 成员

#### dwSize

结构体大小。

#### szFileName

视频文件名。

#### struStartTime

开始时间,参见结构体 NET\_EHOME\_TIME 。

#### struStopTime

结束时间,参见结构体 NET\_EHOME\_TIME 。

#### dwFileSize

视频文件大小。

#### dwFileMainType

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

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

视频文件编号,从0开始。

#### byTimeDiffH

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

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

保留,设为0。

# 15. 1. 130 NET\_EHOME\_REC\_FILE\_COND

视频文件查找条件结构体

# 结构体定义

```
struct{
 DWORD
                    dwChannel:
  DWORD
                    dwRecType:
                    struStartTime;
  NET EHOME TIME
  NET EHOME TIME
                    struStopTime;
  DWORD
                    dwStartIndex;
  DWORD
                    dwMaxFileCountPer;
  BYTE
                    byLocalOrUTC;
                    byRes[63]:
  BYTE
NET EHOME REC FILE COND, *LPNET EHOME REC FILE COND;
```

## 成员

### dwChannel

通道号。

#### dwRecType

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

开始时间,参见结构体 NET EHOME TIME。

## struStopTime

结束时间,参见结构体 NET\_EHOME\_TIME 。

#### dwStartIndex

查询的起始位置,从0开始。

#### dwMaxFileCountPer

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

时间类型: 0-设备本地时间, 即设备 OSD 时间; 1-UTC 时间。

## byRes

保留,设为0。

# 15. 1. 131 NET\_EHOME\_RECORD\_CHAN

关联的录像通道参数结构体

# 结构体定义

```
struct {
  BYTE     byAnalogChanNum;
  BYTE     byAnalogChan[MAX_ANALOG_CHANNUM/*32*/];
  BYTE     byRes1[3];
  WORD     wDigitChanNum;
  BYTE     byDigitChan[MAX_DIGIT_CHANNUM/*480*/];
  BYTE     byRes2[62];
} NET_EHOME_RECORD_CHAN, *LPNET_EHOME_RECORD_CHAN;
```

## 成员

## byAnalogChanNum

模拟通道数量,只读。

## byAnalogChan

模拟通道状态,用数组下标表示: 0-未启用,1-启用。

#### byRes1

保留,设为0。

## wDigitChanNum

数字通道数量,只读。

## byDigitChan

数字通道状态,用数组下标表示,0-未启用,1-启用。

# byRes2

保留,设为0。

# 15. 1. 132 NET\_EHOME\_RELATE\_IPC\_INFO

关联的网络摄像机信息结构体

# 结构体定义

# 成员

## dwIpcNum

关联的网络摄像机数量。

## struIpcInfo

网络摄像机信息,参见结构体 NET\_EHOME\_IPC\_INFO。

# 15. 1. 133 NET\_EHOME\_RELEASE\_SERVER\_INFO

信息发布系统信息结构体

```
struct{
   DWORD dwKeepAliveSeconds;
   BYTE
          byAlarmServerIP[32];
   DWORD dwAlarmServerPort;
   DWORD dwAlarmServerType;
   DWORD dwAlarmServerTcpPort;
   BYTE
          byNtpServerIP[32];
   DWORD
          dwNtpServerPort;
   DWORD dwNtpInterval;
          byPicServerIP[32];
   BYTE
   DWORD dwPicServerPort:
   DWORD dwPicServerType;
           byBlackListAddr[32];
   BYTE
          byBlackListName[32];
   BYTE
   DWORD dwBlackListPort;
          byBlackListUser[32];
   BYTE
   BYTE
          byBlackListPasswd[32];
   DWORD dwTranserialSvrPort;
   BOOL
          bReliableTransmission;
} NET EHOME_RELEASE_SERVER_INFO, *LPNET_EHOME_RELEASE_SERVER_INFO;
```

# dwKeepAliveSeconds

保活时间,单位:秒。

#### bvAlarmServerIP

报警管理服务器的 IP 地址。

#### dwAlarmServerPort

报警管理服务器的端口号。

## dwAlarmServerType

报警管理服务器的通信类型: 0-TCP (默认), 1-UDP。

## dwAlarmServerTcpPort

报警管理服务器下发给终端下发的报警端口号。

## byNtpServerIP

NTP 服务器的 IP 地址。

### dwNtpServerPort

NTP 服务器的端口号。

# dwNtpInterval

NTP 时间同步的间隔。

## byPicServerIP

存储服务器的 IP 地址。

#### dwPicServerPort

存储服务器的端口号。

## dwPicServerType

存储服务器的通信类型: O-TCP (默认), 1-UDP。

## byBlackListAddr

黑名单管理服务器的地址。

#### byBlackListName

黑名单管理服务器的名称。

#### dwBlackListPort

黑名单管理服务器的端口号。

#### byBlackListUser

黑名单管理服务器的用户名。

## byBlackListPasswd

黑名单管理服务器的密码。

#### dwTranserialSvrPort

传输服务器的端口号。

#### bReliableTransmission

是否开启可靠传输: True-是, False-否。

# 15. 1. 134 NET EHOME REMOTE CTRL PARAM

控制参数结构体

## 结构体定义

```
struct{
  DWORD    dwSize;
  void    *lpCondBuffer;

  DWORD    dwCondBufferSize;
  void    *lpInbuffer;

  DWORD    dwInBufferSize;

  BYTE    byRes[32];
} NET_EHOME_REMOTE_CTRL_PARAM, *LPNET_EHOME_REMOTE_CTRL_PARAM;
```

## 成员

#### dwSize

结构体大小。

#### 1pCondBuffer

保存条件参数的缓冲区,由 NET\_ECMS\_RemoteControl 的控制命令(dwCommand)决定,请参见备注中的表格。

#### dwCondBufferSize

条件参数缓冲区大小。

#### 1pInbuffer

保存控制参数的缓冲区,由 NET\_ECMS\_RemoteControl 的(dwCommand)决定,请参见备注中的表格。

## dwInBufferSize

控制参数缓冲区大小。

#### **byRes**

保留,设为0。

备注

dwCommand, lpCondBuffer 和 lpInbuffer 之间的关系请参见下表。

dwCommand	1pCondBuffer	lpInbuffer
NET_EHOME_MANUAL_IOOUT	Nu11	NET_EHOME_MANUAL_IOOUT_CTRL
NET_EHOME_PTZ_CTRL	4 字节通道号 (long)	NET_EHOME_PTZ_PARAM
NET_EHOME_PRESET_CTRL	4 字节通道号 (long)	NET_EHOME_PRESET_PARAM
NET_EHOME_PZIN	4字节通道号 (long)	NET_EHOME_PZIN_PARAM
NET_EHOME_PTRACK	4 字节通道号 (long)	NET_EHOME_POINT

# 15. 1. 135 NET\_EHOME\_REPLACE\_MATERIAL

素材更换参数结构体

# 结构体定义

```
struct{
BYTE szServerIP[NET_EHOME_MAX_TYPE_LEN/*32*/];
DWORD dwServerPort;
DWORD dwMaterialId;
DWORD dwMaterialSeq;
}NET_EHOME_REPLACE_MATERIAL, *LPNET_EHOME_REPLACE_MATERIAL;
```

# 成员

szServerIP

服务器 IP 地址。

dwServerPort

服务器端口号。

dwMaterialId

素材 ID。

dwMaterialSeq

素材编号。

# 15. 1. 136 NET\_EHOME\_SCHEDTIME

时间段参数结构体

# 结构体定义

```
struct{
  BYTE    byStartHour;
  BYTE    byStartMin;
  BYTE    byStopHour;
  BYTE    byStopMin;
}NET_EHOME_ALARM_TIME_COND, *LPNET_EHOME_ALARM_TIME_COND;
```

### 成员

byStartHour

开始时间:时。

byStartMin

开始时间:分。

byStopHour

结束时间:时。

byStopMin

结束时间:分。

# 15. 1. 137 NET\_EHOME\_SCREEN\_SHOT

终端截屏参数结构体

# 结构体定义

```
struct{
  DWORD dwServerPort;
  DWORD dwClientFdIndex;
}NET_EHOME_SCREEN_SHOT, *LPNET_EHOME_SCREEN_SHOT;
```

# 成员

dwServerPort

服务器端口号。

dwClientFdIndex

客户端FD索引号。

# 15. 1. 138 NET\_EHOME\_SCREEN\_SHOT\_EX

终端截屏参数结构体

# 结构体定义

```
struct{
  DWORD dwServerPort;
  DWORD dwClientFdIndex;
  BYTE szKmsURL[256];
} NET_EHOME_SCREEN_SHOT_EX, *LPNET_EHOME_SCREEN_SHOT_EX;
```

## 成员

#### dwServerPort

服务器端口号。

#### dwClientFdIndex

客户端 FD 索引号。

#### szKmsURL

截屏图片在 KMS 服务器上的 URL 地址。

# 15. 1. 139 NET\_EHOME\_SCREEN\_SHOT\_RET

终端截屏响应结构体

# 结构体定义

# 成员

### dwConnfd

终端和服务器之间异步通信的套接字。

#### dwSeq

终端和服务器之间异步通信的 ID。

#### szUUID

截屏图片的 UUID。

#### szPicURL

截屏图片在 KMS 服务器中的 URL 地址。

# 15. 1. 140 NET\_EHOME\_SEND\_PARAM

数据发送参数结构体

# 结构体定义

```
struct{
   DWORD dwSize;
   DWORD dwRecvTimeOut;
   BYTE bySendTimes;
   BYTE byRes2[127];
} NET_EHOME_SEND_PARAM, *LPNET_EHOME_SEND_PARAM;
```

# 成员

#### dwSize

结构体大小。

#### dwRecvTimeOut

接收超时时间,单位: ms。

#### bySendTimes

报文发送次数,在网络较差时可减少丢包的情况。默认为1次,最大3次。

#### byRes2

保留。

# 15. 1. 141 NET\_EHOME\_SERVER\_INFO

服务器信息结构体

# 结构体定义

# ISUPSDK (通用) 开发指南

NET\_EHOME\_IPADDRESS struNTPSever;
DWORD dwNTPInterval;

NET\_EHOME\_IPADDRESS struPictureSever;
DWORD dwPicServerType;

NET EHOME BLACKLIST SEVER struBlackListServer;

BYTE byRes[128];

} NET EHOME SERVER INFO, \*LPNET EHOME SERVER INFO;

## 成员

#### dwSize

结构体大小。

### dwKeepAliveSec

心跳间隔,单位:秒,默认:15秒。

#### dwTimeOutCount

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

报警管理服务器的 TCP 地址,参见结构体 NET\_EHOME\_IPADDRESS。

#### struUDPAlarmSever

报警管理服务器的 UDP 地址,参见结构体 NET EHOME IPADDRESS。

### dwAlarmServerType

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

NTP 服务器地址信息,参见结构体 NET EHOME IPADDRESS。

#### dwNTPInterval

NTP 时间同步的间隔,单位:秒。

#### struPictureSever

存储服务器的地址信息,参见结构体 NET\_EHOME\_IPADDRESS 。

## dwPicServerType

存储服务器(SS)类型: 0-Tomcat, 1-VRB, 2-云存储, 3-KMS, 4-ISUP5.0。

#### struBlackListServer

黑名单管理服务器,参见 NET EHOME BLACKLIST SEVER。

### byRes

保留,设为0。

# 15. 1. 142 NET\_EHOME\_SERVER\_INFO\_V50

服务器信息结构体(V50)

# 结构体定义

```
struct{
                               dwSize;
  DWORD
  DWORD
                               dwKeepAliveSec;
  DWORD
                               dwTimeOutCount:
  NET_EHOME_IPADDRESS
                               struTCPAlarmSever:
  NET EHOME IPADDRESS
                               struUDPAlarmSever;
  DWORD
                               dwAlarmServerType;
  NET EHOME IPADDRESS
                               struNTPSever:
  DWORD
                               dwNTPInterval:
  NET EHOME IPADDRESS
                               struPictureSever;
  DWORD
                               dwPicServerType;
  NET EHOME BLACKLIST SEVER
                               struBlackListServer;
  NET EHOME IPADDRESS
                               struRedirectSever;
                               byClouldAccessKey[64];
  BYTE
  BYTE
                               byClouldSecretKey[64]:
                               byClouldHttps;
  BYTE
                               byRes1[3];
  BYTE
                               dwAlarmKeepAliveSec;
  DWORD
  DWORD
                               dwAlarmTimeOutCount:
  BYTE
                               byRes[372];
}NET_EHOME_SERVER_INFO_V50, *LPNET_EHOME_SERVER_INFO_V50;
```

### 成员

#### dwSize

结构体大小。

#### dwKeepAliveSec

心跳间隔,单位: 秒,默认: 15s。

### dwTimeOutCount

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

报警管理服务器的 TCP 地址,参见结构体 NET EHOME IPADDRESS。

### struUDPAlarmSever

报警管理服务器的 UDP 地址,参见结构体 NET\_EHOME\_IPADDRESS。

#### dwAlarmServerType

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

NTP 服务器地址信息,参见结构体 NET EHOME IPADDRESS。

#### dwNTPInterval

NTP 时间同步的间隔,单位:秒。

#### struPictureSever

存储服务器的地址信息,参见结构体 NET\_EHOME\_IPADDRESS 。

#### dwPicServerType

存储服务器(SS)类型: 0-Tomcat, 1-VRB, 2-云存储, 3-KMS, 4-ISUP5.0。

### struBlackListServer

黑名单管理服务器的参数,参见结构体 NET\_EHOME\_BLACKLIST\_SEVER 。

#### struRedirectSever

重定向服务器的地址,参见结构体 NET EHOME IPADDRESS。

# byClouldAccessKey

云存储的访问密码。

## byClouldSecretKey

云存储的密钥。

# byClouldHttps

启用云存储类型: 1-HTTPS, 0-HTTP。

#### byRes1

保留。

#### dwAlarmKeepAliveSec

报警心跳间隔,单位: 秒,默认: 30s。

#### dwAlarmTimeOutCount

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。

#### **byRes**

保留。

# 15. 1. 143 NET\_EHOME\_SET\_REREGISTER\_MODE

设备注册模式结构体

```
struct{
  DWORD dwSize;
  DWORD dwReRegisterMode;
}NET_EHOME_SET_REREGISTER_MODE, *LPNET_EHOME_SET_REREGISTER_MODE;
```

# 成员

dwSize

结构体大小。

### dwReRegisterMode

是否支持重复注册(仅支持 2.0 版本 ISUP): 0-支持(默认), 1-不支持, 返回错误码 "403"。

# 15. 1. 144 NET\_EHOME\_SHOT\_PIC

弹图命令参数结构体

# 结构体定义

```
struct{
   BYTE    byServerIP[32];
   DWORD   dwServerPort;
   DWORD   dwShotPicID;
}NET_EHOME_SHOT_PIC, *LPNET_EHOME_SHOT_PIC;
```

# 成员

byServerIP

服务器 IP 地址。

dwServerPort

服务器端口号。

dwShotPicID

弹图 ID。

# 15. 1. 145 NET\_EHOME\_SS\_CLIENT\_PARAM

文件上传客户端参数结构体

# 成员

#### enumType

文件上传客户端类型,参见枚举定义、 NET EHOME SS CLIENT TYPE 。

#### struAddress

存储服务器(SS)的 IP 地址。

### byHttps

是否启用 HTTPS 通信。

## byRes

保留,设为0。

# 相关接口

NET ESS CreateClient

# 15. 1. 146 NET\_EHOME\_SS\_LISTEN\_HTTPS\_PARAM

HTTPS 监听参数结构体

# 结构体定义

```
struct{
  BYTE    byHttps;
BYTE    byVerifyMode;
BYTE    byCertificateFileType;
BYTE    byPrivateKeyFileType;
char    szUserCertificateFile[MAX_PATH];
char    szUserPrivateKeyFile[MAX_PATH];
DWORD    dwSSLVersion;
BYTE    byRes3[360];
}NET_EHOME_SS_LISTEN_HTTPS_PARAM, *LPNET_EHOME_SS_LISTEN_HTTPS_PARAM;
```

## 成员

#### byHttps

是否启用 HTTPS: 0-不启用, 1-启用。

## byVerifyMode

校验模式: 0-单项校验。

## byCertificateFileType

证书类型: 0-PEM, 1-ANS1

### byPrivateKeyFileType

私钥类型: 0-PEM, 1-ANS1

#### szUserCertificateFile

公钥证书存储路径。

### szUserPrivateKeyFile

私钥证书存储路径。

#### dwSSLVersion

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器匹配。

### byRes3

保留。

# 15. 1. 147 NET\_EHOME\_SS\_LISTEN\_PARAM

监听参数结构体

# 结构体定义

```
struct{
 NET_EHOME_IPADDRESS
                             struAddress;
                             szKMS_UserName[MAX_KMS_USER_LEN/*512*/];
  char
  char
                             szKMS Password[MAX KMS USER LEN/*512*/];
  EHomeSSStorageCallBack
                             fnSStorageCb;
  EHomeSSMsgCallBack
                             fnSMsgCb;
  char
                             szAccessKey[MAX CLOUD AK SK LEN/*64*/];
  char
                             szSecretKey[MAX_CLOUD_AK_SK_LEN/*64*/];
                            *pUserData;
  void
  BYTE
                             byHttps;
                             byRes1[3];
  BYTE
  EHomeSSRWCallBack
                             fnSSRWCb;
                            byRes[56];
  BYTE
}NET_EHOME_SS_LISTEN_PARAM, *LPNET_EHOME_SS_LISTEN_PARAM;
```

### 成员

#### struAddress

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 NET EHOME IPADDRESS。

#### szKMS UserName

KMS 的用户名。

### szKMS\_Password

KMS 的密码。

#### fnSStorageCb

存储信息的回调函数,参见 EHomeSSStorageCallBack 。

#### fnSMsgCb

存储文件的回调函数,参见 EHomeSSMsgCallBack。

### pUserData

用户参数。

#### byHttps

是否启用 HTTPS 通信: 0-不启用, 1-启用。

#### byRes1

保留。

#### fnSSRWCb

读写的回调函数,参见 EHomeSSRWCallBack。

#### **bvRes**

保留,设为0。

### 备注

读写回调函数( *EHomeSSRWCal | Back* )和存储回调函数( *EHomeSSStorageCal | Back* )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 *EHomeSSRWCal | Back* 生效。

## 相关接口

NET ECMS StartListen

# 15. 1. 148 NET\_EHOME\_SS\_LOCAL\_SDK\_PATH

存储服务器的本地加载路径结构体

```
struct{
  char    sPath[MAX_PATH/*260*/];
  BYTE    byRes[128];
}NET_EHOME_SS_LOCAL_SDK_PATH, *LPNET_EHOME_SS_LOCAL_SDK_PATH;
```

# 成员

sPath

加载路径。

byRes

保留。

# 15. 1. 149 NET\_EHOME\_SS\_TOMCAT\_MSG

Tomcat 服务器信息结构体

# 结构体定义

## 成员

#### szDevUri

```
设备请求 URL,如: "/service/upload_pic.php? puid=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",URL中的字段定义为:
```

#### 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 × 3600 × 100 + 分 × 60 × 100 + 秒 × 100。

#### direction

方向, 计算方式为: 实际方向值(北的方向值为0,单位: degree) × 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。

## 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/0 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。

#### dwPicNum

图片数量。

# pPicURLs

图片 URL。

## byRes

保留,设为0。

# 15. 1. 150 NET\_EHOME\_STOPPLAYBACK\_PARAM

回放停止请求参数结构体

## 结构体定义

# 成员

#### 1SessionID

回放的会话 ID。

#### 1Handle

输出参数指针,在异步模式中作为异步回调的标识,对应 NET\_EHOME\_CMSCB\_DATA 中的 dwHandle。

## **byRes**

保留。

# 15. 1. 151 NET\_EHOME\_STOPSTREAM\_PARAM

预览停止请求参数结构体

# 结构体定义

```
BYTE byRes[120];
```

} NET EHOME STOPSTREAM PARAM, \*LPNET EHOME STOPSTREAM PARAM;

# 成员

#### 1SessionID

预览的会话 ID, 由 NET\_ECMS\_StartGetRealStreamV11 返回。

#### 1Handle

输出参数指针,在异步模式中作为异步回调的标识,对应 NET\_EHOME\_CMSCB\_DATA 中的 dwHandle。

#### **byRes**

保留。

# 15. 1. 152 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

# 结构体定义

# 成员

#### 1SessionID

通过流媒体服务器的语音对讲会话 ID。

#### 1Handle

输出参数句柄,在异步模式中作为异步回调的标识。对应 NET\_EHOME\_CMSCB\_DATA 中的 dwHandle。

#### **b**vRes

保留。

# 15. 1. 153 NET\_EHOME\_SWITCH\_DAILY\_PLAN

每日定时开关机计划参数结构体

# 成员

## dwDayPlanCnt

一天中开关机时间节点个数。

## struDayPlan

开关机参数,参见结构体 NET EHOME SWITCH PLAN 。

# 15. 1. 154 NET\_EHOME\_SWITCH\_PLAN

开关机参数结构

# 结构体定义

```
struct{
  DWORD   dwId;
  DWORD   dwSwitchType;
  DWORD   dwSwitchDate;
  DWORD   dwSwitchTime;
}NET_EHOME_SWITCH_PLAN, *LPNET_EHOME_SWITCHE_PLAN;
```

## 成员

#### dwId

配置节点编号。

### dwSwitchType

开关机:1-开机,2-关机。

#### dwSwitchDate

开关机日期。

#### dwSwitchTime

开关机时间,如 "235959"表示时间为 23:59:59。

# 15. 1. 155 NET\_EHOME\_SWITCH\_PLAN\_PARAM

发送至终端的开关机计划参数结构体

```
struct{
   NET_EHOME_PLAY_SCHEDULE_TYPE1 enumPlanType;
   union
   {
      NET_EHOME_SWITCH_DAILY_PLAN struDailyPlan;
      NET_EHOME_SWITCH_WEEKLY_PLAN struWeeklyPlan;
   }unionSwitchPlan;
} unionSwitchPlan;
} NET_EHOME_SWITCH_PLAN_PARAM, *LPNET_EHOME_SWITCH_PLAN_PARAM;
```

# 成员

### enumPlanType

计划类型: 周计划或日计划,参见枚举 NET EHOME PLAY SCHEDULE TYPE1。

#### unionSwitchPlan

struDailyPlan

日计划参数,参见结构体 NET\_EHOME\_SWITCH\_DAILY\_PLAN 。

#### struWeeklyPlan

周计划参数,参见结构体 NET\_EHOME\_SWITCH\_WEEKLY\_PLAN 。

# 15. 1. 156 NET\_EHOME\_SWITCH\_WEEKLY\_PLAN

开关机周计划参数结构体

## 结构体定义

### 成员

#### dwWeekPlanCnt

一周中配置了开关机计划的天数。

#### struWeekPlan

开关机周计划参数,参见结构体 NET\_EHOME\_SWITCHE\_DAY\_OF\_WEEK\_PLAN 。

# 15. 1. 157 NET\_EHOME\_SWITCHE\_DAY\_OF\_WEEK\_PLAN

开关机周计划参数结构体

# 成员

#### dwId

周内星期 ID。

### enumDayOfWeek

星期,参见枚举 NET\_EHOME\_DAY\_OF\_WEEK1。

### struDayOfWeekPlan

开关机日计划参数,参见结构体 NET\_EHOME\_SWITCH\_DAILY\_PLAN 。

# 15. 1. 158 NET EHOME SYSCOMPONENT REDUCED ADDR

终端组件信息结构体

# 结构体定义

# 成员

#### szComponentName

组件名称,如"SDK"。

## szComponentPkgName

组件包名称,如 "com. hikvision. sdk"。

#### szComponentVersion

组件版本信息,如 "v2.0.2 build 20180705"。

#### dwIsEnableAutoStart

是否开机自启动。

### dwIsSupportUpgrade

是否支持远程升级。

### **byRes**

保留。

# 15. 1. 159 NET\_EHOME\_TERM\_DEFAULT\_GATEWAY

终端网关参数结构体

# 结构体定义

```
struct{
  BYTE     szIpAddress[32];
  BYTE     szIpv6Address[32];
}NET_EHOME_TERM_DEFAULT_GATEWAY, *LPNET_EHOME_TERM_DEFAULT_GATEWAY;
```

## 成员

## szIpAddress

IPv4 地址的网关。

### szIpv6Address

IPv6 地址的网关。

# 15. 1. 160 NET\_EHOME\_TERM\_IP\_ADDRESS

终端 IP 地址参数结构体

# 结构体定义

# 成员

## szIpVersion

IP 地址版本: v4 和 v6。

## szIpAddress

IP 地址。

#### szSubnetMask

子网掩码。

### szIpv6Address

IPv6地址,目前暂不支持。

#### szBitMask

IPv6地址的网关。

#### strDefaultGateway

终端的网关参数,参见结构体 NET\_EHOME\_TERM\_DEFAULT\_GATEWAY。

# 15. 1. 161 NET\_EHOME\_TERM\_TIME\_ZONE

终端时区信息结构体

# 结构体定义

```
struct {
    BYTE szTimeZone[32];
}NET_EHOME_TERM_TIME_ZONE, *LPNET_EHOME_TERM_TIME_ZONE;
```

# 成员

### szTimeZone

时区,如GMT+08。

# 15. 1. 162 NET\_EHOME\_TERMINAL\_CONTROL

终端控制参数结构体

# 结构体定义

# 成员

dwControlType

控制命令类型,参见枚举 NET\_EHOME\_CONTROL\_TYPE 。

#### struInsertInfo

插播参数,参见结构体 NET EHOME INSERT INFO。

#### struPlanCtrl

信息发布终端的定时计划控制参数,参见结构体 NET\_EHOME\_PLANCTRLPARAM。

#### dwConnPort

数据传输端口,保留。

# 15. 1. 163 NET\_EHOME\_TERMINAL\_CONTROL\_V20

终端控制参数结构体(V20)

# 结构体定义

### 成员

#### dwControlType

控制命令类型,参见枚举 NET\_EHOME\_CONTROL\_TYPE 。

#### struInsertInfo

插播参数,参见结构体 NET EHOME INSERT INFO V20。

#### struPlanCtrl

信息发布终端的定时计划控制参数,参见结构体 NET\_EHOME\_PLANCTRLPARAM。

#### dwConnPort

数据传输端口,保留。

#### **bvRes**

保留。

# 15. 1. 164 NET\_EHOME\_TERMINAL\_INFO

终端信息结构体

```
struct{
  NET EHOME IDENTIFICATION struDevIdentify;
  DWORD
                            dwNetUintType;
  BYTE
                            byDeviceID[NET_EHOME_MAX_TYPE_LEN/*32*/];
  BYTE
                            byPassWord[NET EHOME MAX TYPE LEN/*32*/];
  BYTE
                            byFirmWareVersion[NET EHOME MAX TYPE LEN/*32*/];
                            byLocalIP[NET_EHOME_MAX_TYPE_LEN/*32*/];
  BYTE
  DWORD
                            dwLocalPort;
  DWORD
                            dwDevType;
  DWORD
                            dwManufacture;
  BOOL
                            bDetectPackage;
  BOOL
                            bReliableTrans;
                            byServerUserName[NET EHOME MAX NAME LEN/*32*/];
  BYTE
                            byServerUserPassword[NET_EHOME_MAX_NAME_LEN/*32*/];
  BYTE
  BYTE
                            byTerminalName[NET_EHOME_MAX_NAME_LEN/*32*/];
} NET EHOME TERMINAL INFO, *LPNET EHOME TERMINAL INFO;
```

# 成员

### struDevIdentify

终端 ID, 参见结构体 NET EHOME IDENTIFICATION。

### dwNetUintType

网络类型。

#### byDeviceID

设备 ID。

#### byPassWord

密码。

#### byFirmWareVersion

固件版本信息。

#### byLocal IP

终端的通信 IP 地址。

#### dwLocalPort

本地端口号。

### dwDevType

设备类型。

#### dwManufacture

厂商: 0-海康威视。

### bDetectPackage

封装类型: 1-探测包, 0-注册包。

#### bReliableTrans

是否支持可靠传输: 1-是, 0-否。

### byServerUserName

服务器用户名。

### byServerUserPassword

服务器密码。

### byTerminalName

待注册的终端名称。

# 15. 1. 165 NET\_EHOME\_TERMINAL\_NAME

终端名称结构体

# 结构体定义

### 成员

### szTermName

终端名称。

# 15. 1. 166 NET\_EHOME\_TERMINAL\_PROGRESS

终端进度信息结构体

## 结构体定义

```
struct{
   DWORD   dwTerminalId;
   DWORD   dwTermStatus;
   DWORD   dwProgessType;
   DWORD   dwUniqueID;
   BYTE   byMainProgess;
   BYTE   bySubProgess;
}NET_EHOME_TERMINAL_PROGRESS, *LPNET_EHOME_TERMINAL_PROGRESS;
```

# 成员

### dwTerminalId

终端 ID。

#### dwTermStatus

终端状态:1-正常,2-异常。

### dwProgessType

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

日程 ID, 节目 ID, 或素材 ID, 升级时该字段无效。

#### **byMainProgess**

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。

### bySubProgess

单个文件的下载进度。

# 15. 1. 167 NET\_EHOME\_TERMINAL\_REPORT\_INFO

终端上报数据结构体

# 结构体定义

```
struct {
   DWORD    dwSubCmd;
   DWORD    dwContentLen;
   BYTE    byContentBuf[2048];
   BYTE    byRes[64];
} NET_EHOME_TERMINAL_REPORT_INFO, *LPNET_EHOME_TERMINAL_REPORT_INFO;
```

# 成员

### dwSubCmd

标识数据内容的类型。

#### dwContentLen

数据内容的长度。

#### byContentBuf

储存数据内容的缓冲区。

#### **byRes**

保留。

# 15. 1. 168 NET\_EHOME\_TERMINAL\_STATE

终端状态结构体

# 结构体定义

# 成员

# enumPlayState

播放状态,参见枚举 NET\_EHOME\_TERMINAL\_PLAY\_STATE 。

### enumTerminalType

终端类型,参见枚举 NET\_EHOME\_TERMINAL\_TYPE 。

#### enumInsertState

插播状态,参见枚举 NET\_EHOME\_INSERT\_STATE 。

## bySoftVersion

终端的软件版本号。

#### byLocal IP

终端的 IP 地址。

## dwTermAbnormalState

终端是否处于温度异常关机状态: 0-否, 1-是。

# 15. 1. 169 NET EHOME TERMINAL UPG ST REPORT

终端返回的升级状态结构体

# 结构体定义

```
DWORD dwRetVal;
DWORD dwUpgradPercent;
} NET_EHOME_TERMINAL_UPG_ST_REPORT, *LPNET_EHOME_TERMINAL_UPG_ST_REPORT;
```

# 成员

#### dwLength

结构体大小。

#### dwCheckSum

校验和。

#### dwRetVal

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

升级进度(百分比)。

# 15. 1. 170 NET\_EHOME\_TEXT\_MESSAGE

文字消息内容结构体

# 结构体定义

# 成员

### dwId

文字消息 ID。

### szMsgName

消息名称。

#### szContent

消息内容。

### szBeginTime

插播开始时间。

### szEndTime

插播结束时间。

# 15. 1. 171 NET\_EHOME\_TIME

时间信息结构体

# 结构体定义

```
\mathtt{struct}\,\{
  WORD
           wYear;
  BYTE
          byMonth;
  BYTE
           byDay;
          byHour;
  BYTE
  BYTE
          byMinute;
          bySecond;
  BYTE
  BYTE
          byRes1;
          wMSecond;
  WORD
          byRes2[2];
  BYTE
}NET_EHOME_PREVIEWINFO_IN, *LPNET_EHOME_PREVIEWINFO_IN;
```

# 成员

wYear

年

byMonth

月

byDay

日

byHour

时

byMinute

分

bySecond

秒

byRes1

保留。

wMSecond

毫秒

### byRes2

保留。

# 15. 1. 172 NET\_EHOME\_TIME\_ADJUST\_PARAM

时间同步参数结构体

# 结构体定义

```
struct {
  DWORD dwTimeZone;
  BYTE szTime[32];
}NET_EHOME_TIME_ADJUST_PARAM, *LPNET_EHOME_TIME_ADJUST_PARAM;
```

# 成员

dwTimeZone

时区,参见枚举 NET\_EHOME\_TIME\_ZONE。

szTime

时间信息,格式为YYYY-MM-DD HH:MM:SS (年-月-日 时:分:秒)。

# 15. 1. 173 NET EHOME UPGRADE CONN PARAM

TCP 链路升级命令参数结构体

# 结构体定义

```
struct{
BYTE szServerIP[32];
DWORD dwServerPort;
}NET_EHOME_UPGRADE_CONN_PARAM, *LPNET_EHOME_UPGRADE_CONN_PARAM;
```

# 成员

szServerIP

服务器 IP 地址。

dwServerPort

服务器端口号。

# 15. 1. 174 NET\_EHOME\_VERSION\_INFO

版本信息结构体

# 成员

dwSize

结构体大小。

sSoftwareVersion

软件版本号。

sDSPSoftwareVersion

编码版本号。

sPanelVersion

面板版本号。

sHardwareVersion

硬件版本号。

# 15. 1. 175 NET\_EHOME\_VOICE\_TALK\_IN

语音对讲请求的输入参数结构体

## 结构体定义

## 成员

dwVoiceChan

语音对讲的通道号。

struStreamSever

流媒体服务器的地址信息,参见结构体 NET\_EHOME\_IPADDRESS。

## byRes

保留,设为0。

# 15. 1. 176 NET\_EHOME\_VOICE\_TALK\_OUT

语音对讲请求的输出参数结构体

# 结构体定义

# 成员

#### 1SessionID

语音对讲请求的会话 ID,由设备返回。

## byRes

保留,设为0。

# 15. 1. 177 NET\_EHOME\_VOICETALK\_DATA

用于转发的语音数据结构体

# 结构体定义

```
struct {
  BYTE *pSendBuf;
  DWORD dwDataLen;
  BYTE byRes[128];
}NET_EHOME_VOICETALK_DATA, *LPNET_EHOME_VOICETALK_DATA;
```

# 成员

### pSendBuf

语音数据缓冲区。

### dwDataLen

语音数据大小。

### byRes

保留,设为0。

# 15. 1. 178 NET\_EHOME\_VOICETALK\_DATA\_CB\_INFO

语音对讲回调数据结构体

# 结构体定义

```
struct {
    BYTE *pData;
    DWORD dwDataLen;
    BYTE byRes[128];
} NET_EHOME_VOICETALK_DATA_CB_INFO, *LPNET_EHOME_VOICETALK_DATA_CB_INFO;
```

# 成员

pData

保存语音数据的缓冲区。

dwDataLen

保存语音数据的缓冲区大小。

**byRes** 

保留。

# 15. 1. 179 NET\_EHOME\_VOICETALK\_DATA\_CB\_PARAM

语音对讲数据回调参数的结构体

# 结构体定义

# 成员

fnVoiceTalkDataCB

语音对讲数据的回调函数,参见 VOICETALK\_DATA\_CB。

pUserData

用户参数。

**byRes** 

保留,设为0。

# 15. 1. 180 NET\_EHOME\_VOICETALK\_NEWLINK\_CB\_INFO

语音对讲回调信息结构体

# 结构体定义

## 成员

### szDeviceID

设备 ID。

#### dwEncodeType

语音编码类型: 0-0ggVorbis, 1-G711U, 2-G711A, 3-G726, 4-AAC, 5-MP2L2, 6-PCM。

#### sDeviceSerial

设备序列号。

#### dwAudioChan

语音对讲通道号。

#### 1SessionID

语音对讲的会话 ID, 由设备返回。

#### bvToken

令牌。

#### **byRes**

保留,设为0。

# 15. 1. 181 NET\_EHOME\_VOICETALK\_PARA

语音对讲或语音转发参数结构体

# 结构体定义

struct {		
BOOL	bNeedCBNoEncData;	

# ISUPSDK(通用) 开发指南

```
fVoiceDataCallBack
                        cbVoiceDataCallBack;
  DWORD
                        dwEncodeType;
                        *pUser;
  void
  BYTE
                        byVoiceTalk;
                        byDevAudioEnc;
  BYTE
  BYTE
                        byRes1[2];
                        lHandle;
 LONG
                        byRes[56];
  BYTE
}NET EHOME VOICETALK PARA, *LPNET EHOME VOICETALK PARA;
```

## 成员

#### bNeedCBNoEncData

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

音频数据的回调函数,参见 fVoiceDataCallBack。

#### dwEncodeType

支持的语音编码类型: 0-OggVorbis, 1-G. 711U, 2-G. 711A, 3-G. 726, 4-AAC, 5-MP2L2, 6-PCM。

#### pUser

用户参数。

### byVoiceTalk

工作模式: 0-语音对讲, 1-语音转发。

### byDevAudioEnc

设备的音频编码类型: 0-OggVorbis, 1-G. 711U, 2-G. 711A, 3-G. 726, 4-AAC, 5-MP2L2, 6-PCM。

#### byRes1

保留,设为0。

#### 1Handle

语音对讲异步回调的消息句柄。

#### **byRes**

保留,设为0。

# 15. 1. 182 NET EHOME VOLUME DAILY PLAN

音量调整日计划结构体

# 成员

## dwDayPlanCnt

一天中音量调节的时间节点数量。

## struDayPlan

音量调节日计划参数,参见结构体 NET EHOME VOLUME PLAN。

# 15. 1. 183 NET\_EHOME\_VOLUME\_DAY\_OF\_WEEK\_PLAN

音量调节周计划参数结构体

# 结构体定义

### 成员

#### dwId

周内星期 ID。

### enumDayOfWeek

星期,参见枚举 NET\_EHOME\_DAY\_OF\_WEEK1。

#### struDayOfWeekPlan

音量调节日计划参数,参见结构体 NET EHOME VOLUME DAILY PLAN

# 15. 1. 184 NET\_EHOME\_VOLUME\_PLAN

音量参数结构体

# 结构体定义

```
struct{
    DWORD    dwId;
```

```
DWORD dwVolumeBeginDate;

DWORD dwVolumeBeginTime;

DWORD dwVolumeEndDate;

DWORD dwVolumeEndTime;

DWORD dwVolumeEndTime;

NET_EHOME_VOLUME_PLAN, *LPNET_EHOME_VOLUME_PLAN;
```

# 成员

#### dwId

音量调节 ID。

#### dwVolumeValue

音量值。

#### dwVolumeBeginDate

音量调节起始日期。

#### dwVolumeBeginTime

音量调节开始时间,如"235959"表示时间 23:59:59。

# dwVolumeEndDate

音量调节结束日期。

#### dwVolumeEndTime

音量调节结束时间。

### 15. 1. 185 NET EHOME VOLUME PLAN PARAM

需发送至终端的音量调节参数结构体

# 结构体定义

```
struct{
  NET_EHOME_PLAY_SCHEDULE_TYPE1 enumPlanType;
  union {
    NET_EHOME_VOLUME_DAILY_PLAN struDailyPlan;
    NET_EHOME_VOLUME_WEEKLY_PLAN struWeeklyPlan;
  }unionVolumePlan;
} NET_EHOME_VOLUME_PLAN_PARAM, *LPNET_EHOME_VOLUME_PLAN_PARAM;
```

### 成员

#### enumPlanType

计划类型,参见枚举 NET\_EHOME\_PLAY\_SCHEDULE\_TYPE1 unionVolumePlan

音量调节计划联合体,详见以下参数。

#### struDailyPlan

日计划,参见 NET EHOME VOLUME DAILY PLAN 。

#### struWeeklyPlan

周计划,参见 NET EHOME VOLUME WEEKLY PLAN 。

### 15. 1. 186 NET EHOME VOLUME WEEKLY PLAN

音量调节周计划参数结构体

## 结构体定义

### 成员

#### dwWeekPlanCnt

一周中有音量调节任务的天数。

#### struWeekPlan

音量调节周计划参数,参见结构体 NET\_EHOME\_VOLUME\_DAY\_OF\_WEEK\_PLAN 。

### 15. 1. 187 NET\_EHOME\_WEATHER\_INFO

天气信息结构体

## 结构体定义

```
struct{
   BYTE byCityCode[32];
   BYTE byWeatherData[4 * 1024];
} NET_EHOME_WEATHER_INFO, *LPNET_EHOME_WEATHER_INFO;
```

#### 成员

byCityCode

城市代码。

#### byWeatherData

天气数据。

## 15. 1. 188 NET\_EHOME\_WIRELESS\_INFO\_CFG

无线参数结构体

### 结构体定义

```
struct {
   DWORD    dwSize;
   DWORD    dwInfoTransInterval;
   BYTE    byEnable;
   BYTE    byRes[47];
} NET_EHOME_WIRELESS_INFO_CFG, *LPNET_EHOME_WIRELESS_INFO_CFG;
```

### 成员

dwSize

结构体大小。

#### dwInfoTransInterval

上传时间间隔,单位:秒。

byEnable

是否支持: 0-否, 1-是。

**byRes** 

保留。

## 15. 1. 189 NET\_EHOME\_XML\_CFG

传输配置参数结构体

## 结构体定义

```
struct{
  void
           *pCmdBuf;
  DWORD
           dwCmdLen;
           *pInBuf:
  void
  DWORD
           dwInSize;
           *pOutBuf;
  void
  DWORD
           dwOutSize;
  DWORD
           dwSendTimeOut;
           dwRecvTimeOut;
  DWORD
           *pStatusBuf;
  void
           dwStatusSize;
  DWORD
  BYTE
           byRes[24];
}NET_EHOME_XML_CFG, *LPNET_EHOME_XML_CFG;
```

## 成员

### pCmdBuf

字符串格式的命令。

#### dwCmdLen

命令的字符数量。

#### pInBuf

输入参数。

#### dwInSize

输入缓冲区大小。

## pOutBuf

输出参数。

#### dwOutSize

输出缓冲区大小。

#### dwSendTimeOut

数据发送超时时间,单位:毫秒,默认:5s。

#### dwRecvTimeOut

数据接收超时时间,单位:毫秒,默认:5s。

#### pStatusBuf

保存返回状态信息(XML 格式)的缓冲区;若不需要可设为"空"。

#### dwStatusSize

状态缓冲区的大小。

#### **byRes**

保留,设为0。

### 相关接口

NET\_ECMS\_XMLConfig

## 15. 1. 190 NET\_EHOME\_XML\_REMOTE\_CTRL\_PARAM

传输参数结构体

## 结构体定义

 $\mathtt{struct}\,\{$ 

DWORD **dwSize**;

```
*lpInbuffer;
 void
 DWORD
           dwInBufferSize;
 DWORD
           dwSendTimeOut:
 DWORD
           dwRecvTimeOut;
           *lpOutBuffer;
 void
 DWORD
           dwOutBufferSize;
           *lpStatusBuffer;
 void
 DWORD
           dwStatusBufferSize;
 BYTE
           byRes[16];
NET EHOME XML REMOTE CTRL PARAM, *LPNET EHOME XML REMOTE CTRL PARAM;
```

## 成员

#### dwSize

结构体大小。

### 1pInbuffer

保存输入参数的缓冲区,由传输的 ISUP 控制命令决定。

#### dwInBufferSize

输入缓冲区大小。

#### dwSendTimeOut

数据发送超时时间,单位:毫秒,默认:5 s.

#### dwRecvTimeOut

数据接收超时时间,单位:毫秒,默认:5 s.

#### 1pOutBuffer

输出缓冲区。

#### dwOutBufferSize

输出缓冲区大小。

#### 1pStatusBuffer

状态缓冲区, 若不需要该参数, 可设为"空"。

#### dwStatusBufferSize

状态缓冲区的大小。

#### **byRes**

保留,设为0。

### 相关接口

NET\_ECMS\_XMLRemoteControl

## 15. 1. 191 NET\_EHOME\_ZONE

区域参数结构体

## 结构体定义

```
struct{
  DWORD   dwX;
  DWORD   dwY;
  DWORD   dwWidth;
  DWORD   dwHeight;
}NET_EHOME_ZONE, *LPNET_EHOME_ZONE;
```

### 成员

#### dwX

X轴坐标。

#### dwY

Y轴坐标。

#### dwWidth

区域宽度。

#### dwHeight

区域高度。

## 15.2 枚举定义

## 15. 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,
                                       = 7,
 ALARM_TYPE_VIDEO_COVERED
 ALARM_TYPE_MOTION
                                       = 8,
 ALARM TYPE STANDARD NOTMATCH
                                       = 9,
 ALARM_TYPE_SPEEDLIMIT_EXCEED
                                       = 10,
 ALARM_TYPE_PIR
                                       = 11,
 ALARM_TYPE_WIRELESS
                                       = 12,
```

## ISUPSDK(通用) 开发指南

```
ALARM TYPE CALL HELP
                                     = 13.
ALARM TYPE DISARM
                                    = 14.
ALARM_TYPE_PIC_UPLOAD_FAIL
ALARM_TYPE_LOCAL_BRG
ALARM TYPE STREAM PRIVATE
                                   = 15.
                                   = 16,
                                   = 17,
ALARM_TYPE_LOCAL_REC_EXCEPTION
                                   = 18,
ALARM TYPE UPGRADE FAIL
ALARM TYPE ILLEGAL ACCESS
                                    = 19.
ALARM_TYPE_SOUNDLIMIT_EXCEED
ALARM_TYPE_TRIFFIC_VIOLATION
                                    = 80,
                                   = 90,
ALARM TYPE ALARM CONTROL
                                   = 96,
                              = 97,
= 98,
= 99,
ALARM TYPE FACE DETECTION
ALARM TYPE DEFOUSE DETECTION
ALARM TYPE AUDIO EXCEPTION
                                   = 100,
ALARM TYPE SCENE CHANGE
                               = 101,
ALARM_TYPE_TRAVERSE_PLANE
ALARM TYPE ENTER AREA
                                   = 102,
ALARM TYPE LEAVE AREA
                                   = 103,
ALARM TYPE INTRUSION
                                   = 104.
ALARM TYPE LOITER
                                   = 105,
                                   = 106,
ALARM TYPE LEFT TAKE
ALARM TYPE CAR STOP
                                    = 107,
ALARM TYPE MOVE FAST
                                    = 108,
                                   = 109,
ALARM TYPE HIGH DENSITY
ALARM TYPE PDC BY TIME
                                   = 110.
                                   = 111.
ALARM TYPE PDC BY FRAME
ALARM TYPE LEFT
                                   = 112.
                                   = 113,
ALARM TYPE TAKE
                                   = 114,
ALARM TYPE ROLLOVER
ALARM TYPE COLLISION
                                   = 115.
                                   = 256,
ALARM TYPE FLOW OVERRUN
                                  = 257,
ALARM TYPE WARN FLOW OVERRUN
ALARM TYPE DEV CHANGED STATUS
                                   = 700,
ALARM TYPE CHAN CHANGED STATUS
                                   = 701,
                                   = 702,
ALARM TYPE HD CHANGED STATUS
                                   = 703,
ALARM TYPE DEV TIMING STATUS
ALARM_TYPE_CHAN_TIMING_STATUS
                                   = 704,
ALARM_TYPE_HD_TIMING_STATUS
ALARM_TYPE_RECORD_ARNORMAL
                                    = 705.
                                  = 706,
ALARM TYPE ENV LIMIT
                                   = 8800,
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.
                            = 40971,
ALARM TYPE LEFT CROSS LINE
```

## ISUPSDK(通用) 开发指南

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

## 15. 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。

## 15. 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。

## 15. 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。

## 15. 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。

## 15. 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_CALLBACK_TYPE;
```

### 成员

NET\_EHOME\_REALSTREAM\_CB

预览异步回调。

NET EHOME PLAYBACK CB

回放异步回调。

NET\_EHOME\_VOICETALK\_CB

语音对讲回调。

## 15. 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 liberypto.so)所在路径。

NET\_EHOME\_CMS\_INIT\_CSSLEAY\_PATH

设置 OpenSSL 的通信库(ssleay32.dll or libssl.so)所在路径

## 15. 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。

## 15. 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

## 15. 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。

#### NET\_EHOME\_EALARM\_INIT\_CFG\_SSLEAY\_PATH

设置 OpenSSL 的通信库 (ssleay32.dll or libssl.so) 所在路径

## 15. 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

## 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。

### 15. 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_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。

### NET EHOME ESTREAM INIT CFG SSLEAY PATH

设置 OpenSSL 的通信库 (i.e., ssleay32.dll or libssl.so)的路径。

## 15. 2. 13 NET EHOME INSERT STATE

枚举插播状态。

### 枚举定义

```
typedef enum{
  NET_EHOME_INVAILD_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\_INVAILD\_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。

## 15. 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。

## 15. 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,
                              = 1,
 AMS_ADDRESS
                              = 2,
 SEND PARAM
 SET_REREGISTER_MODE
                              = 3,
 LOCAL_CFG_TYPE_GENERAL
                              = 4,
 COM_PATH
                              = 5,
 SESSIONKEY REQ MOD
                              = 6
}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。

## 15. 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。

 ${\tt NET\_EHOMe\_CONTROL\_INPUTPLAY}$ 

是否启用定时输入计划。

## 15. 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。

## 15. 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。

## 15. 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。

## 15. 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。

## 15. 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。

## 15. 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。

## 15. 2. 23 NET\_EHOME\_REGISTER\_TYPE

枚举回调数据类型。

### 枚举定义

```
enum _NET_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
  ENUM_EHOME50_DEV_DAS_REQ
                                =5,
  ENUM_DEV_SESSIONKEY_REQ
                                =6,
                                =7,
  ENUM DEV DAS REREGISTER
  ENUM DEV DAS EHOMEKEY ERROR =8
}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 EHOMEKEY ERROR

校验密码失败信息。

## 15. 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
```

### 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

## 15. 2. 25 NET\_EHOME\_SS\_INIT\_CFG\_TYPE

枚举初始化配置类型。

#### 枚举定义

#### **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 加密库的路径。

NET EHOME SS INIT CFG SSLEAY PATH

设置 OpenSSL 通信库的路径。

## 15. 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

#### 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。

#### 15. 2. 27 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。

## 15. 2. 28 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。

## 15. 2. 29 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。

## 15. 2. 30 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。

## 15. 2. 31 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。
```

## 15. 2. 32 NET\_EHOME\_TRANSMINT\_PROTOCAL

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_PROTOCAL;
```

## 成员

```
NET_EHOME_TCP
TCP
NET_EHOME_UDP
UDP
NET_EHOME_MCAST
MCAST
```

## 15. 2. 33 SEARCH\_TYPE

枚举查找类型。

## 枚举定义

## 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

# 第16章 命令和报文

## 16.1 控制命令报文

Enter a short description of your concept here (optional).

This is the start of your concept.

#### 16. 1. 1 ADJUSTTIME

手动设备校时。

## 请求报文

```
<Pre>
<Pre>
<Pre>
<Pre>
<Pre>
<Pre>
<Pre>
<Pre>
```

## 响应报文

#### 16. 1. 2 GETDEVICEWORKSTATUS

查找并获取设备工作状态。

## 请求报文

### 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-->
     \langle Run \rangle 0 \langle /Run \rangle
     <!--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>O</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
 <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

### 16. 1. 4 REBOOT

重启设备。

## 请求报文

## 响应报文

#### 16. 1. 5 RESTARTIVSLIB

重启算法库。

### 请求报文

## 响应报文

#### 16. 1. 6 SETDEFENSESTATUS

远程布防或撤防。

## 请求报文

```
</Params>
</PPVSPMessage>
```

### 响应报文

## 16. 1. 7 UPDATE

升级设备。

### 请求报文

```
<?xml version="1.0" encoding="GB2312" ?>
<PPVSPMessage>
 <Version>2.5
 <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

```
<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

Enter a short description of your concept here (optional).

This is the start of your concept.

### 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〉
《!—通道号,从1开始—〉
《/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〉
```

```
《ConfigXML》
《ALARMTIME》<! --布防计划中的时间段--》
《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》
《StartTime》00:00</StartTime》
《StopTime》00:00</StopTime》
《/TIME》
《/ALARMTIME》
《/ConfigXML》
《/Params》
```

### 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)

```
⟨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)

```
<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)

#### 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)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0</Version>
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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)

### 输出参数(p0utBuf)

```
<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)

### 输出参数(pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0</Version>
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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)

### 输出参数(p0utBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0</Version>
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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>
```

#### 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
```

```
<Params>
<ConfigCmd>GetHideAlarmPara</ConfigCmd>
<!--通道号,从1开始-->
<ConfigParam1></ConfigParam1>
</Params>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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>
```

```
<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>
```

#### 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>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0</Version>
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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>
```

#### 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>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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)

### 输出参数(p0utBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0
 <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
 <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

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0</Version>
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <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>
       <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)

## 输出参数(pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <WhichCommand>GETDEVICECONFIG</WhichCommand>
 <Status>200</Status>
 <Description>OK</Description>
 <Params>
   <ConfigXML>
     <RS232INF0>
       <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)

# 输出参数(pOutBuf)

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0
 <Sequence>1</Sequence>
 <CommandType>RESPONSE
 <WhichCommand>GETDEVICECONFIG</WhichCommand>
 <Status>200</Status>
 <Description>OK</Description>
 <Params>
   <ConfigXML>
     <RS485INF0>
       <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⟩</p>
       <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>
     </RS232INFO>
   </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

```
<Params>
<ConfigCmd>GetVILostPara</ConfigCmd>
<!--通道号,从1开始-->
```

```
<ConfigParam1></ConfigParam1>
</Params>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<PPVSPMessage>
 <Version>4.0
 <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

```
<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⟩</p>
     </TIME>
    <TIME>
```

### 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

```
<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-->
 </ConfigParaml>
 <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-->
```

```
</pre
```

### 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

```
<Params>
 <ConfigCmd>SetAlarmTriggerCapture
 <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-->
 </ConfigParaml>
 <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>
```

```
《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》
```

# 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

```
《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》
《ConfigParam2》
《ConfigNML》
《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》
《/RECORDCHAN》
《/ConfigXML》
```

### 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-->
 </ConfigParaml>
 <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>
```

### 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>
```

#### 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---〉
《ConfigParaml〉</ConfigParaml〉
《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(p0utBuf)

#### 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)

### 输出参数(p0utBuf)

#### 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

```
《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〉
《AREATOP〉<!--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〉
《AREARECT〉
```

```
<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>
```

#### 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

```
</ConfigXML>
</Params>
```

#### 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

```
<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>
      <!--<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>
```

### 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)

### 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-->
 </ConfigParaml>
 <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>
```

### 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)

### 输出参数(p0utBuf)

# 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

```
<Params>
<ConfigCmd>SetPassengerPara</ConfigCmd>
<ConfigParam1>
```

#### 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

```
<Params>
 <ConfigCmd>SetRecordPlanPara</ConfigCmd>
 <ConfigParam1>
   <!--通道号,从1开始-->
 </ConfigParaml>
 <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-->
        <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>
```

#### 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

#### 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

```
<Params>
 <ConfigCmd>SetRS485Para</ConfigCmd>
 《ConfigParam1》<!--通道号,从1开始--></ConfigParam1>
 <ConfigXML>
   <RS485INF0>
     <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〉〈1--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><1--解码器地址,取值范围为0到255--></DecoderAddr>
   </RS232INFO>
```

```
</ConfigXML>
```

#### 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)

# 16.3 XML 或 JSON 报文

Enter a short description of your concept here (optional). This is the start of your concept.

## 16. 3. 1 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. 3. 2 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)

```
《"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*/
"channel ID":,
/*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 "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*/
   "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, IS08601 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" */
   "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*/
     "ethnic": {
/*可选项,是否为少数民族*/
      "confidence":
/*可选项, float, 置信度*/
      "value": ""
/*可选项, string, 是否少数民族: no-否, yes-是, 最大长度为 32 字符*/
     "mask": {
/*可选项,是否带口罩*/
      "confidence":
/*可选项, float, 置信度*/
      "value": ""
/*可选项, string, 是否带口罩: 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*/
     "beard": {
      "value": ""
/*可选项, string, 是否有胡子: no-否, yes-是, unknown-未知, 最大长度为 32 字符*/
     "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*/
/*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, IS08601 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, IS08601 时间格式, 最佳抓拍的离开时间, 如 "2004-05-03T17:30:08+08:00" */
   }],
   "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/ison
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*/
 "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*/
   "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*/
    "ethnic": {
/*可选项,是否为少数民族*/
      "confidence":,
/*可选项, float, 置信度*/
/*可选项, string, 是否少数民族: no-否, yes-是, 最大长度为 32 字符*/
    },
     "mask": {
/*可选项,是否带口罩*/
      "confidence":
```

```
/*可选项, float, 置信度*/
      "value": ""
/*可选项, string, 是否带口罩: no-否, ves-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*/
    },
     "beard": {
      "value": ""
/*可选项, string, 是否有胡子: no-否, yes-是, unknown-未知, 最大长度为 32 字符*/
     "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*/
/*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, IS08601 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, IS08601 时间格式, 最佳抓拍的离开时间, 如 "2004-05-03T17:30:08+08:00" */
   }],
   "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-
```

## 16.3.3 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,
2018070420504791100E40Ryu67pug56nUQ8C6uof48mz97s61mk13P7C7Hp9u2*/
   "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*/
/*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": "beard",
      "value": ""
/*可选项, string, 是否有胡子: "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,
2018070420504791100E40Ryu67pug56nUQ8C6uof48mz97s61mk13P7C7Hp9u2*/
   "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*/
/*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",
/*可选项, string, 是否微笑: "no"-否, "yes"-是; 最大取值长度为 32 字符*/
      "description": "beard",
      "value": ""
```

```
/*可选项, string, 是否有胡子: "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*/
/*必选项, float, 人脸框左上角的 X 坐标*/
/*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="F4F665D6A18E41308CE9934DCDDD11111";
filename="faceCapturePicture.jpg"
Content-Type:image/jpeg
Content-Length: 118579
Content-ID: faceImage
```

```
<B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—
```

#### 16.3.4 JSON\_EventNotificationAlert\_FacePicComparisonMsg

关于人脸图片比对报警详情的 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 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*/
 "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=7a57a5a7ffffffffc1a0316369671314*/
```

```
"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*/
    "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, IS08601 格式的抓拍时间,最大长度为 64 字符,例如,2016-09-10 00:00:00:00.000*/
    "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*/
/*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://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":
     "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":,
```

```
"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": "",
        "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*/
```

## 16.3.5 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*/
 "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*/
```

## 16.3.6 JSON\_EventNotificationAlert\_MultiTargetTypeDetectMsg

关于混合目标检测事件详情的 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"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*/
 "deviceID": "",
/*可选项, string, 设备 ID, 在 ISUP SDK 传输过程中, 该节点必选*/
 "CaptureResult": [{
/*抓拍和侦测结果*/
   "targetID": ,
/*必选项,int,抓拍到的目标 ID*/
   "Face": {
/*可选项,人脸属性*/
    "Rect": {
/*可选项,人脸小图框信息*/
      "height":,
/*必选项, float, 框的高度*/
      "width":,
/*必选项, float, 框的宽度*/
      "x": ,
/*必选项, float, 框左上角 X 坐标*/
      "y":
/*必选项, float, 框左上角 Y 坐标*/
     "Property": [{
/*人脸属性详情*/
      "description": "",
      "value": ""
    }],
     "contentID1": "",
/*人脸小图*/
    "contentID2": "",
```

```
/*背景图片*/
     "FacePictureRect": {
/*可选项,人脸小图框信息*/
"height": ,
/*必选项, float, 框的高度*/
      "width":,
/*必选项, float, 框的宽度*/
     "x": ,
/*必选项, float, 框左上角 X 坐标*/
/*必选项, float, 框左上角 Y 坐标*/
   "Human": {
/*可选项,人体属性*/
     "Rect": {
/*可选项,人体小图的框信息*/
      "height":,
/*必选项, float, 框的高度*/
     "width":,
/*必选项, float, 框的宽度*/
"x": ,
/*必选项, float, 框左上角的 X 坐标*/
/*必选项, float type, 框左上角的 Y 坐标*/
     "Property": [{
/*人体属性详情*/
      "description": "",
      "value": ""
     }],
     "modeldata": "",
/*可选项,由 Base64 编码的人体建模数据*/
     "contentID1": "",
/*人体小图*/
    "contentID2": ""
/*背景图片*/
   "NonMotor": {
/*可选项,非机动车属性*/
    "Rect": {
/*可选项, 非机动车小图框信息*/
     "height":,
/*必选项, float, 框的高度*/
     "width":,
/*必选项, float, 框的宽度*/
     "x":,
/*必选项, float, 框左上角 X 坐标*/
    ″y″:
/*必选项, float, 框左上角 Y 坐标*/
   "Property": [{
```

```
/*非机动车属性详情*/
     "description": "",
     "value": ""
   }],
   "contentID": ""
/*非机动车小图*/
 "Vehicle": {
/*可选项,车辆属性*/
   "VehicleRect": {
/*可选项,车辆图片框信息*/
"height": ,
/*必选项, float, 框的高度*/
     "width":,
/*必选项, float, 框的宽度*/
/*必选项, float, 框左上角 X 坐标*/
     "y":
/*必选项, float, 框左上角 Y 坐标*/
   "PlateRect": {
/*可选项,车牌图片框信息*/
     "height":,
/*必选项, float, 框的高度*/
     "width":,
/*必选项, float, 框的宽度*/
     "x":,
/*必选项, float, 框左上角 X 坐标*/
/*必选项, float, 框左上角 Y 坐标*/
   "Property": [{
/*车辆属性详情*/
     "description": "",
     "value": ""
   }],
   "contentID1": "",
/*车辆图片*/
   "contentID2": "",
/*车牌背景图片*/
   "contentID3": ""
/*车辆背景图片*/
 "AbsoluteHigh":{
/*PTZ 绝对位置*/
   "elevation":,
/*可选项, int, 上下角度, 取值范围从-900 到 2700*/
   "azimuth":,
/*可选项, int, 左右角度, 取值范围从 0 到 3600*/
  "absoluteZoom":
/*可选项, int, 缩放倍率, 取值范围从 1 到 1000*/
},
```

```
"GSP": {
/*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: [-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: [-180.000000, 180.000000]*/
  "uid":""
/*混合目标检测报警 ID, 其最大取值长度为 64 位,由时间(精确到毫秒)和随机数字组成*/
 "behaviorLinkageUid":""
/*智能事件或报警上传 ID, 其最大取值长度为 64 位, 由时间(精确到毫秒)和随机数字组成*/
--MIME boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD11111";
filename="faceCapturePicture.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: faceImage
{\tt fefefwageegfqaeg} \cdots
--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="F4F665D6A18E41308CE9934DCDDD11111";
filename=" humanImage.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: humanImage
fefefwageegfqaeg...
--MIME boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD11111";
filename=" humanBackgroundImage.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: humanBackgroundImage
fefefwageegfqaeg...
--MIME boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD11111";
```

```
filename="nonMotorImage.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: nonMotorImage
fefefwageegfqaeg...
--MIME boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD11111";
filename="vehicleImage.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: vehicleImage
fefefwageegfqaeg \cdots
--MIME boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD11111";
filename="plateImage.jpg";
Content-Type: image/jpeg
Content-Length: 5798
Content-ID: plateImage
fefefwageegfqaeg...
--MIME boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD11111";
filename=" vehicleBackgroundImage.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: vehicleBackgroundImage
{\tt fefefwageegfqaeg} \cdots
--MIME boundary
Content-Disposition: form-data;
name="F4F665D6A18E41308CE9934DCDDD11111";
filename=" faceLibImage.jpg";
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: faceLibImage
fefefwageegfqaeg...
--MIME boundary--
```

```
/*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 "mixedTargetDetection" */
 "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*/
 "CaptureResult": [{
/*抓拍和侦测结果*/
   "targetID": ,
/*必选项, int, 抓拍到的目标 ID*/
   "Face": {
/*可选项,人脸属性*/
    "Rect": {
/*可选项,人脸小图框信息*/
      "height":,
/*必选项, float, 框的高度*/
      "width": ,
/*必选项, float, 框的宽度*/
      "x": ,
/*必选项, float, 框左上角 X 坐标*/
      "y":
/*必选项, float, 框左上角 Y 坐标*/
     "Property": [{
/*人脸属性详情*/
      "description": "",
      "value": ""
     }],
     "faceImageURL": "",
/*人脸图片 URL, 当上传图片至云存储服务器的时候会自动生成*/
     "faceBackgroundImageURL": ""
/*人脸背景图片 URL, 当上传图片至云存储服务器的时候会自动生成*/
   "Human": {
/*可选项, 人体属性*/
    "Rect": {
/*可选项,人体小图的框信息*/
```

```
"height":,
/*必选项, float, 框的高度*/
      "width":,
/*必选项, float, 框的宽度*/
      "x":,
/*必选项, float, 框左上角的 X 坐标*/
/*必选项, float type, 框左上角的 Y 坐标*/
    "Property": [{
/*人体属性详情*/
      "description": "",
      "value": ""
    }],
    "modeldata": "",
/*可选项,由 Base64 编码的人体建模数据*/
    "humanImageURL": "",
/*可选项,人体图片 URL,当上传图片至云存储服务器的时候会自动生成*/
    "humanBackgroundImageURL": ""
/*可选项,人体背景图片 URL,当上传图片至云存储服务器的时候会自动生成*/
   "NonMotor": {
/*可选项,非机动车属性*/
    "Rect": {
/*可选项, 非机动车小图框信息*/
"height": ,
/*必选项, float, 框的高度*/
    "width": ,
/*必选项, float, 框的宽度*/
    "x": ,
/*必选项, float, 框左上角 X 坐标*/
/*必选项, float, 框左上角 Y 坐标*/
   "Property": [{
/*非机动车属性详情*/
    "description": "",
    "value": ""
   "nonMotorImageURL": "",
/*可选项,非机动车图片,当上传图片至云存储服务器的时候会自动生成*/
   "nonMotorBackgroundImageURL": ""
/*可选项,非机动车背景图片,当上传图片至云存储服务器的时候会自动生成*/
 "Vehicle": {
/*可选项,车辆属性*/
   "VehicleRect": {
/*可选项,车辆图片框信息*/
    "height":,
/*必选项, float, 框的高度*/
    "width":,
/*必选项, float, 框的宽度*/
```

```
"x":,
/*必选项, float, 框左上角 X 坐标*/
/*必选项, float, 框左上角 Y 坐标*/
   "PlateRect": {
/*可选项,车牌图片框信息*/
    "height":,
/*必选项, float, 框的高度*/
    "width":,
/*必选项, float, 框的宽度*/
    "x":,
/*必选项, float, 框左上角 X 坐标*/
/*必选项, float, 框左上角 Y 坐标*/
   "Property": [{
/*车辆属性详情*/
    "description": "",
    "value": ""
   }],
   "vehicleImageURL": "",
/*车辆图片,当上传图片至云存储服务器的时候会自动生成*/
   "plateImageURL": "",
/*车牌图片, 当上传图片至云存储服务器的时候会自动生成*/
   "vehicleBackgroundImageURL": ""
/*车辆背景图片, 当上传图片至云存储服务器的时候会自动生成*/
 "AbsoluteHigh": {
/*PTZ 绝对位置*/
   "elevation":,
/*可选项, int, 上下角度, 取值范围从-900 到 2700*/
   "azimuth":,
/*可选项, int, 左右角度, 取值范围从 0 到 3600*/
   "absoluteZoom":
/*可选项, int, 缩放倍率, 取值范围从1到1000*/
 "GSP": {
/*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: [-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: [-180.000000, 180.000000]*/
 "uid":"",
/*混合目标检测报警 ID, 其最大取值长度为 64 位,由时间(精确到毫秒)和随机数字组成*/
 "behaviorLinkageUid":""
/*智能事件或报警上传 ID, 其最大取值长度为 64 位, 由时间(精确到毫秒)和随机数字组成*/
```

## 16.3.7 JSON\_EventNotificationAlert\_PeopleQueuingUpMsg

关于区域人数检测事件详情的 ISON 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 "personQueueCounting" */
 "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*/
 "RegionCapture" {
/*抓拍区域信息*/
   "humanCounting": {
/*人员数量*/
    "count":
/*必选项, int, 实际人员数量, 其取值范围从 0 到 60*/
   "rule": {
/*规则属性*/
    "ruleID": ,
/*必选项, int, 规则 ID, 与目标人员区域 ID 对应*/
     "alarmCount": ,
/*必选项, int, 区域人员数量阈值, 其取值范围从 0 到 60*/
     "regionColor": "",
```

```
/*必选项, string, 规则区域颜色: "blue"-蓝色, "red"-红色, "vellow"-黄色*/
     "Region": [{
/*必选项,多边形规则区域坐标(包含3到10条边)*/
/*必选项, float, X 坐标, 其取值范围从 0.000 到 1*/
/*必选项, float, X 坐标, 其取值范围从 0.000 到 1*/
    } ]
    "countTriggerType": "",
/*可选项, string, 区域人数统计报警的触发类型: "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,"range"-人数范围在最大人数(maxAlarmCount)和
报警阈值(AlarmCount)之间*/
    "durationTime":
/*可选项, int, 人数保持时长, 其取值范围从 0 到 3600, 默认时长为 300, 单位: 秒; 当实际保持时长等于该
设置保持时长时,报警触发*/
 "contentID": ""
/*背景图片*/
--MIME boundary--
Content-Type: image/jpeg
Content-Length: 42569
Content-ID: human image
fefefwageegfqaeg...
--MIME boundary-
```

```
"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 "personQueueCounting" */
 "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*/
 "RegionCapture" {
/*抓拍区域信息*/
   "humanCounting": {
/*人员数量*/
    "count":
/*必选项, int, 实际人员数量, 其取值范围从 0 到 60*/
   "rule":{
/*规则属性*/
    "ruleID": ,
/*必选项,int,规则 ID,与目标人员区域 ID 对应*/
    "alarmCount":
/*必选项, int, 区域人员数量阈值, 其取值范围从 0 到 60*/
    "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)*/
/*必选项, float, X 坐标, 其取值范围从 0.000 到 1*/
/*必选项, float, X 坐标, 其取值范围从 0.000 到 1*/
    } ]
    "countTriggerType": "",
/*可选项, string, 区域人数统计报警的触发类型: "greater"-高于人数阈值, "less"-低于人数阈值,
"equal"-等于人数阈值,"unequal"-不等于人数阈值,"range"-人数范围在最大人数(maxAlarmCount)和
人数阈值(AlarmCount)之间*/
    "durationTime":
/*可选项,int,人数保持时长,其取值范围从 0 到 3600,默认时长为 300,单位:秒;当实际保持时长等于该
设置保持时长时,报警触发*/
 "backgroundImageURL": ""
/*背景图片 URL*/
```

## 16.3.8 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

```
"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*/
/*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":,
```

```
"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":,
```

## 16.3.9 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*/
/*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)*/
/*必选项, float, X 坐标, 其取值范围从 0.000 到 1*/
/*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--
```

```
"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*/
/*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*/
/*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*/
```

#### 16.3.10 XML EventNotificationAlert AlarmInputMsg

关于报警输入报警详情的 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--></ipAddress>
 <ipv6Address><!--依赖项, xs:string--></ipv6Address>
 <portNo><!--可选项, xs:integer--></portNo>
 <protocolType><!--可选项, xs:string, 协议类型: HTTP, HTTPS-->
 <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>
 <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"IO"--></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>
 <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〉〈!一必选项, xs:string, 事件或报警推送详情一〉〈/eventPush〉
```

```
</Extensions>
</EventNotificationAlert>
```

#### 16.3.11 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--></ipAddress>
 <ipv6Address><!--依赖项, xs:string--></ipv6Address>
 <portNo><!--可选项, xs:integer--></portNo>
  protocol><!--可选项, xs:string, 协议类型: HTTP, HTTPS-->
 <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>
 <dateTime><!--必选项, xs:datetime, ISO 8601 时间格式, 例如, 2017-04-22T15:39:01+08:00--></
 <activePostCount><!--必选项, xs:integer--></activePostCount>
 <eventType><!--必选项, xs:string, 事件类型: 此处需要设为 "ANPR" -->
 <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>
 <anpr>><!--依赖项,该节点只有当<eventType>为 "ANPR" --></a>
   <region><!--可选项, xs:string, 识别区域--></region>
   <country><!--可选项, xs:string, 国家代码--></country>
   censePlate><!--必选项, xs:string, 车牌号码, 例如, 123456--></licensePlate>
   -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>
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>
   clicenseBright>
     <!--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-->
 \langle /_{\rm X} \rangle
 <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-->
</ri>
<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-->
</ri>
<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>
<1abe1>
 <!--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-->
   </pdys>
   <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-->
   </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-->
   </re>
   <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-->
   </barrierGateCtrlTvpe>
   <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-->
     </re>
     <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-->
     </re>
     <vehileSubLogoRecog>
      <!--可选项, xs:integer, 车辆子品牌-->
     </re>
     <vehileModel>
      <!--可选项, xs:integer, 车辆子品牌的型号-->
     </re>
     <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)-->
     </re>
   </re>
   <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><!--rea-->
     <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-->
      <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-->
       </re>
      <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>
        <!一依赖项,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-->
        \langle \chi \rangle
          <!--必选项, xs:integer, 边框左上角的 X 坐标-->
        \langle X \rangle
        <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>
      <vehice1Rect>
        <!一依赖项,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-->
        \langle / Y \rangle
        <width>
         <!--必选项, xs:integer, 边框的宽度-->
        </width>
          <!--必选项, xs:integer, 边框的高度-->
        </height>
      </re>
      <pictureURL>
        <!--依赖项, xs:string, 图片 URL, 该节点只有在〈dataType〉为"URL"时有效-->
      </pictureURL>
     </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>
 </ANPR>
 <UUID>
   <!--可选项, xs:string, 通用 ID, 用于关联跨服务器抓拍-->
 </UUID>
 <picNum>
   <!--可选项, xs:integer, 图片数量-->
 </picNum>
 <monitoringSiteID>
   <!--可选项, xs:string, 相机编号-->
 </monitoringSiteID>
 <deviceID>
   <!--可选项, xs:string, 设备 ID-->
 </deviceID>
 <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>
```

#### 16. 3. 12 XML\_EventNotificationAlert\_AudioExceptionMsg

关于音频异常报警详情的 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>
 <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 "audioexception" --></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>
```

#### 16.3.13 XML EventNotificationAlert DefocusDetectionMsg

关于虑焦侦测报警详情的 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>
 cprotocolType><!--可选项, xs:string, 协议类型: HTTP, HTTPS-->
 <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>
 <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 "defocus" --></eventType>
 <eventState><!--必选项,xs:string,事件状态(持续事件):active-已发生,inactive-未发生--></
 <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. 3. 14 XML\_EventNotificationAlert\_FaceDetectionMsg

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>
  <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>
```

#### 16.3.15 XML EventNotificationAlert FastMovingDetectionMsg

关于快速移动侦测报警详情的 XML 报文

```
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,如果无报警图片,该节点将不返回--
></detectionPicturesNumber>
 〈bkgUrl〉<!--可选项, xs:string, 背景图片 URL--></bkgUrl>
 <ur><ur>URLCertificationType<!--可选项,xs:string,图片URL认证方式:,no-无认证,digest-摘要认证--></ur></ur>
URLCertificationType>
 <pId><!-可选项, xs:string, 图片 ID--></pId>
</EventNotificationAlert>
```

#### 16.3.16 XML\_EventNotificationAlert\_HDDErrorAlarmMsg

关于硬盘错误报警详情的 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>
 cprotocolType><!--可选项, xs:string, 协议类型: HTTP, HTTPS-->
 <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
 <dvnChannelID><!--可选项, xs:string, 数字通道 ID--></dvnChannelID>
 <channelID><!--依赖项, xs:string, 通道 ID--></channelID>
 <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 "diskerror" --></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>
 <diskNo><!--可选项, xs:integer, 硬盘号--></diskNo>
 <HDDList><!一依赖项, xs:integer, 硬盘列表, 该节点只有在<eventType>为 "diskfull" 或 "diskerror"
```

#### 16.3.17 XML EventNotificationAlert IllegalLoginAlarmMsg

关于非法登录报警详情的 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>
 <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 "illaccess" --></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>
</EventNotificationAlert>
```

#### 16.3.18 XML EventNotificationAlert IntrusionMsg

关于入侵侦测报警详情的 XML 报文

```
<eventState><!--必选项, xs:string, 事件状态(持续事件); active-已发生, inactive-未发生-->//
 <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>
 <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>
 <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,如果无报警图片,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>
</EventNotificationAlert>
```

#### 16.3.19 XML\_EventNotificationAlert\_LineCrossingMsg

关于越界侦测报警详情的 XML 报文

```
<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>
     <detectionTarget><!--可选项, xs:string, 目标类型: human-人, vehicle-车, others-其他--></
detectionTarget>
     <TargetRect><!--可选项-->
   <X><!--必选项, xs:float; 目标边框的 X 坐标, 取值范围: [0,1]--></X>
   < Y><!--必选项, xs:float; 目标边框的 Y 坐标, 取值范围: [0,1]--></ Y>
   <width><!--必选项, xs:float; 目标边框的宽度, 取值范围: [0,1]--></width>
   <height><!--必选项,xs:float;目标边框的高度,取值范围:[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, 事件或报警推送详情--></eventPush>
 <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,如果无报警图片,该节点将不返回--
></detectionPicturesNumber>
 ⟨bkgUrl⟩<!--可选项, xs:string, 背景图片 URL--></bkgUrl>
 <ur><ur>URLCertificationType<!--可选项,xs:string,图片URL认证方式:,no-无认证,digest-摘要认证--</li></ur></ur>
URLCertificationType>
 <pId><!-可选项, xs:string, 图片 ID--></pId>
</EventNotificationAlert>
```

#### 16. 3. 20 XML\_EventNotificationAlert\_LoiteringDetectionMsg

关于徘徊侦测报警详情的 XML 报文

```
<dateTime><!--必选项, xs:datetime, ISO 8601 时间格式,例如,2017-04-22T15:39:01+08:00--></
 <activePostCount><!--必选项, xs:integer--></activePostCount>
 <eventType><!--必选项, xs:string, 事件类型: 此处需要设为 "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,如果无报警图片,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>
</EventNotificationAlert>
```

#### 16. 3. 21 XML\_EventNotificationAlert\_MotionDetectionMsg

关于移动侦测报警详情的 XML 报文

```
<eventDescription><!--必选项, xs:string--></eventDescription>
 <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>
 <deviceID><!--可选项, xs:string, 设备 ID--></deviceID>
   <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. 3. 22 XML EventNotificationAlert ObjectRemovalDetectionMsg

关于物品拿取侦测报警详情的 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-->
 <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>
 <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 "attendedBaggage" --></eventType>
 <eventState><!--必选项, xs:string, 事件状态(持续事件): active-己发生, 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>
《detectionPictureTransType>
《detectionPicturesNumber><!--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--></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>
《/EventNotificationAlert》
```

#### 16. 3. 23 XML\_EventNotificationAlert\_ParkingDetectionMsg

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>
 <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>
 <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, 如果无报警图片, 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> 〈/EventNotificationAlert〉
```

#### 16. 3. 24 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>
 <<!--可选项, xs:string, "HTTP, HTTPS"-->
 <macAddress><!--可选项, xs:string, MAC 地址--></macAddress>
 <dvnChannelID><!--可选项, xs:string, 数字通道号--></dynChannelID>
 <channelID><!--依赖项, xs:string--></channelID>
 <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>
```

#### 16.3.25 XML\_EventNotificationAlert\_PeopleGatheringDetectionMsg

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>
 <<!--可选项, xs:string, 协议类型: HTTP, HTTPS-->
 <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>
 <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>
 <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,如果无报警图片,该节点将不返回--
></detectionPicturesNumber>
 <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>
</EventNotificationAlert>
```

#### 16.3.26 XML\_EventNotificationAlert\_PIRAlarmMsg

关于被动红外报警详情的 XML 报文

#### 16. 3. 27 XML EventNotificationAlert RegionEntranceMsg

关于进入区域侦测报警详情的 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>
 cprotocolType><!--可选项, xs:string, 协议类型: HTTP, HTTPS-->
 <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>
 <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"regionEntrance"--></eventType>
 <eventState><!--必选项,xs:string,事件状态(持续事件): active-已发生,inactive-未发生--></
 <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>
     <detectionTarget><!--可选项, xs:string, 目标类型: human-人, vehicle-车, others-其他--></
detectionTarget>
     <TargetRect><!--可选项-->
   <X><!--必选项, xs:float; 目标边框的 X 坐标, 取值范围: [0,1]--></X>
```

```
< Y><!--必选项, xs:float; 目标边框的 Y 坐标, 取值范围: [0,1]--></ Y>
   <width><!--必选项, xs:float; 目标边框的宽度, 取值范围: [0,1]--></width>
   <height><!--必选项, xs:float; 目标边框的高度, 取值范围: [0,1]--></height>
     </TargetRect>
   </DetectionRegionEntry>
 </DetectionRegionList>
 <channelName><!--可选项, xs:string, 通道名称--></channelName>
 <detectionPictureTransType><!--可选项,xs:string,图片传输方式: "url", "binary" --></
detectionPictureTransType>
 <detectionPicturesNumber><!--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--
></detectionPicturesNumber>
 <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>
</EventNotificationAlert>
```

#### 16.3.28 XML EventNotificationAlert RegionExitingMsg

关于离开区域侦测报警详情的 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>
 <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>
 <dateTime><!--必选项, xs:datetime, ISO 8601时间格式,例如,2017-04-22T15:39:01+08:00--></
 <activePostCount><!--必选项, xs:integer--></activePostCount>
 <eventType><!--必选项, xs:string, 事件类型: 此处需要设为 "regionExiting" --></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>
     <detectionTarget><!--可选项, xs:string, 目标类型: human-人, vehicle-车, others-其他--></
detectionTarget>
```

```
⟨TargetRect⟩⟨!--可选项-->
   <X><!--必选项, xs:float; 目标边框的 X 坐标, 取值范围: [0,1]--></X>
   <Y><!--必选项, xs:float; 目标边框的 Y 坐标, 取值范围: [0,1]--></Y>
   <width><!--必选项, xs:float; 目标边框的宽度, 取值范围: [0,1]--></width>
   <height><!--必选项, xs:float; 目标边框的高度, 取值范围: [0,1]--></height>
     </TargetRect>
   </DetectionRegionEntry>
 </DetectionRegionList>
 <channelName><!--可选项, xs:string, 通道名称--></channelName>
 <detectionPictureTransType><!--可选项,xs:string,图片传输方式:"url","binary"--></
detectionPictureTransType>
 <detectionPicturesNumber><!--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--
></detectionPicturesNumber>
 <br/>
<br/>
⟨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>
</EventNotificationAlert>
```

#### 16. 3. 29 XML\_EventNotificationAlert\_SceneChangeDetectionMsg

关于场景变更侦测报警详情的 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-->
 <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>
 <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 "scenechangedetection" --></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; 事件或报警推送内容--></eventPush>
 </Extensions>
</EventNotificationAlert>
```

#### 16.3.30 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, 协议类型: HTTP, HTTPS-->
 <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>
 <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 "TMA" -->/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>
 <DetectionRegionList>
   <DetectionRegionEntry><!--列表-->
     <regionID><!--必选项, xs:string, 侦测区域 ID--></regionID>
     〈sensitivityLevel〉<!一可选项, 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>
     <TMA>
       <thermometryUnit><!--可选项,xs:string--></thermometryUnit>
       <ruleTemperature><!--可选项, xs:integer--></ruleTemperature>
       <currTemperature><!--可选项, xs:integer--></currTemperature>
       <ruleCalibType><!--可选项, xs:string--></ruleCalibType>
       <ruleType><!--可选项, xs:string--></ruleType>
       <MaximumTemperaturePoint>
         <RegionCoordinates><!--列表-->
           <positionX><!--可选项, xs:integer--></positionX>
           <positionY><!--可选项, xs:integer--></positionY>
         </RegionCoordinates>
       </MaximumTemperaturePoint>
       <AbsoluteHigh>
         <elevation><!--可选项, xs:float--></elevation>
```

```
<azimuth><!--可选项, xs:float--></azimuth>
         <absoluteZoom><!--可选项, xs:float--></absoluteZoom>
       </AbsoluteHigh>
       cpresetNo><!--可选项, xs:integer, 预置点号--></presetNo>
     </TMA>
   </DetectionRegionEntry>
 </DetectionRegionList>
 <channelName><!--可选项, xs:string--></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,热成像附加信息数据URL--></thermalInfoURL>
 <URLCertificationType>
   <!--可选项,xs:string,图片 URL 认证方式: ,no-无认证(针对武汉云存储 URL),digest-摘要认证(针
对 DVR 或 NVR 的本地存储 URL), 最大长度为 32 字节-->
 </URLCertificationType>
</EventNotificationAlert>
--boundary
Content-Disposition: form-data;
Content-Type: image/pjpeg
Content-Length: 47432
〈图片数据〉
.....JFIF.....
--boundary--
```

#### 示例

Enter an example to illustrate your reference here (optional).

### 16. 3. 31 XML\_EventNotificationAlert\_UnattendedBaggageDetectionMsg

关于物品遗留侦测报警详情的 XML 报文

```
<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,如果无报警图片,该节点将不返回--
></detectionPicturesNumber>
 <ur><ur>URLCertificationType<!--可选项,xs:string,图片URL认证方式:,no-无认证,digest-摘要认证--</li></ur></ur>
URLCertificationType>
 <pId><!-可选项, xs:string, 图片 ID--></pId>
</EventNotificationAlert>
```

#### 16.3.32 XML EventNotificationAlert VideoLossMsg

关于视频丢失报警详情的 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>
 <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>
 <dateTime><!--必选项, xs:datetime, ISO 8601时间格式,例如,2017-04-22T15:39:01+08:00--></
 <activePostCount><!--必选项, xs:integer--></activePostCount>
 <eventType><!--必选项, xs:string, 事件类型: 此处需要设为 "videoloss" --></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; 事件或报警推送内容--></eventPush>
```

```
</Extensions>
</EventNotificationAlert>
```

#### 16. 3. 33 XML EventNotificationAlert VideoTamperingMsg

关于视频遮挡侦测报警详情的 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>
 colType><!--可选项, xs:string, 协议类型: HTTP, HTTPS-->
 <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>
 <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 "shelteralarm" --></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; 事件或报警推送内容--></eventPush>
 </Extensions>
</EventNotificationAlert>
```

# 附录 A. 附录

### A.1 事件类型和详情

表 A-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	XML_EventNotificationAlert_Moti onDetectionMsg
视频遮挡	shelteralarm	XML_EventNotificationAlert_Vide oTamperingMsg
视频丢失	videoloss	XML_EventNotificationAlert_Vide oLossMsg
报警输入	10	XML_EventNotificationAlert_Alar mInputMsg
音频异常侦测	audioexception	XML_EventNotificationAlert_AudioExceptionMsg
虚焦侦测	defocus	XML_EventNotificationAlert_Defo cusDetectionMsg
场景变更侦测	scenechangedetection	XML_EventNotificationAlert_Scen eChangeDetectionMsg
人脸侦测	facedetection	XML_EventNotificationAlert_Face DetectionMsg
入侵侦测	fielddetection	XML_EventNotificationAlert_IntrusionMsg
越界侦测	linedetection	XML_EventNotificationAlert_Line CrossingMsg
进入区域侦测	regionEntrance	XML_EventNotificationAlert_RegionEntranceMsg

事件类型	eventType 取值	pAlarmData (事件报文)
离开区域侦测	regionExiting	XML_EventNotificationAlert_RegionExitingMsg
徘徊侦测	loitering	XML_EventNotificationAlert_Loit eringDetectionMsg
人员聚集侦测	group	XML_EventNotificationAlert_Peop leGatheringDetectionMsg
快速移动侦测	rapidMove	XML_EventNotificationAlert_Fast MovingDetectionMsg
停车侦测	parking	XML_EventNotificationAlert_Park ingDetectionMsg
物品遗留侦测	unattendedBaggage	XML_EventNotificationAlert_Unat tendedBaggageDetectionMsg
物品拿取侦测	attendedBaggage	XML_EventNotificationAlert_ObjectRemovalDetectionMsg
被动红外报警	PIR	XML_EventNotificationAlert_PIRA larmMsg
混合目标检测	mixedTargetDetection	JSON_EventNotificationAlert_Mul tiTargetTypeDetectMsg
人脸抓拍	faceCapture	JSON_EventNotificationAlert_Fac eCaptureMsg
排队时长检测	personQueueTime	JSON_EventNotificationAlert_Wai tingTimeDetectionMsg
区域排队人数检测	personQueueCounting	JSON_EventNotificationAlert_PeopleQueuingUpMsg
人脸图片比对结果	alarmResult	JSON_EventNotificationAlert_Fac ePicComparisonMsg
人脸建模结果	faceSnapModeling	JSON_EventNotificationAlert_Fac eModelingMsg

事件类型	eventType 取值	pAlarmData (事件报文)
车辆检测(自动车牌识别)	ANPR	
非法摆摊检测	cityManagement	JSON_EventNotificationAlert_Unr egisteredStreetVendorMsg
客流统计报告	PeopleCounting	XML_EventNotificationAlert_PeopleCountingEventMsg
热度图报告	heatmap	JSON_EventNotificationAlert_Hea tMapMsg
温度报警	ТМА	XML_EventNotificationAlert_Temp eratureAlarmMsg
硬盘错误报警	diskerror	XML_EventNotificationAlert_HDDE rrorAlarmMsg
非法登录报警	illaccess	XML_EventNotificationAlert_Ille galLoginAlarmMsg

### A. 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。
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。

错误名	错误码	描述
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。
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。

错误名	错误码	描述
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。
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 ss1eay32. d11 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_PLAYBACK_ERR_TYPE_ UNSUPPORTED	3501	该回放类型不支持。

错误名	错误码	描述
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。
NET_SS_CLIENT_ERR_CLOUD_ BESTNODE_FAIL	3604	获取最优节点失败。

### 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_ NOTSUPORT	601	不支持。
NET_AUDIOINTECOM_ERR_ ALLOC_MEMERY	602	申请内存失败。
NET_AUDIOINTECOM_ERR_ PARAMETER	603	参数值不正确。

错误名	错误码	描述
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。
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。

### A.3 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
0x41	缓冲区溢出

值	描述
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
0x4002	风扇异常

值	描述
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
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)

值	描述
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)
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)

值	描述
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)
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)

值	描述
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)
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)

值	描述
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)
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)

值	描述
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
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

值	描述
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
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)

值	描述
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
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)

值	描述
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)
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)

值	描述
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)
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

值	描述
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)
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

#### A. 3.1 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

日志类型	描述
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
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

日志类型	描述
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
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
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

日志类型	描述
0x18	GPRS 恢复
0x19	3G 网络异常
0x1a	3G 网络恢复
0x1b	SIM 卡异常
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
0x17	添加 DVR/NVR 操作用户

日志类型	描述
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
0x2e	远程: 开始录像
0x2f	远程:停止录像

日志类型	描述
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
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

日志类型	描述
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
0x5c	远程: 关闭警灯
0xf0	本地: 激活设备
0xf1	远程: 激活设备

日志类型	描述
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

### A. 4 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

产品类型及系列	产品型号	加密索引	RS-485
	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
	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	支持
	DS-2CD7047EWD	DS-2CD7047EWD	支持

产品类型及系列	产品型号	加密索引	RS-485
	DS-2CD7067EWD	DS-2CD7067EWD	支持
	DS-2CD70C7EWD-A(国内标配)(白)	DS-2CD70C7EWD	支持
	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
	DS-2CD7A47EWD-XZ(S) (2.8-12mm) (8~32mm)	DS-2CD7A47EWD-X	支持

产品类型及系列	产品型号	加密索引	RS-485
	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
	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

产品类型及系列	产品型号	加密索引	RS-485
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
	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

产品类型及系列	产品型号	加密索引	RS-485
	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
	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

产品类型及系列	产品型号	加密索引	RS-485
	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
	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

产品类型及系列	产品型号	加密索引	RS-485
	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
	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)	/	/

产品类型及系列	产品型号	加密索引	RS-485
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

### A. 5 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-18
DS-9616N-I8(8*8T AI 盘)
DS-9632N-18
DS-9616N-I8
DS-9608N-18
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)

产品型号
DS-8632N-18/24P
DS-8616N-I8
DS-FN32CGX (B)
DS-8608N-I8 (8*6T)
DS-8616N-I8 (8*6T)
DS-8632N-18
DS-8664N-I8
DS-FN08CGX (B)
DS-8664N-I16
DS-FN08MG
DS-8616N-I16
DS-8632N-I8 (8*6T)
DS-8608N-18
DS-8664N-18/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-18/MB-84
DS-FN32DG(C)

产品型号
DS-8616N-I8/MB-84
DS-8664N-I8 (8*6T)
DS-GJZ9N08-F86 (B)
DS-FN32MG
DS-8608N-18/MB-84
DS-8616N-I16 (16*6T)
DS-8664N-I16 (16*6T)
DS-GJZ9N08-F96 (B)
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-12
DS-7616N-I2
DS-7708N-I4/8P/1G
DS-7616N-I2/16P
DS-7608N-12/8P
DS-7632N-I2/16P
DS-7632N-12

