



ISUPSDK (General)

[Development Guide](#)

Legal Notices

About this document

- This document is only used as a guide for related products and may differ from the actual product. Please refer to the actual product.

The Company does not provide any express or implied statement or warranty for this document. • We recommend that you use this document under the guidance of professionals.

Trademark Notice

The trademarks mentioned in this manual are owned by their respective owners.

Disclaimer

- To the maximum extent permitted by law, this document and the products described (including its hardware, software, firmware, etc.) are provided "as is" and may contain defects or errors. The company does not provide any form of express or implied warranty, including but not limited to merchantability, quality satisfaction, fitness for a particular purpose, etc.; nor will it compensate for any special, incidental, incidental or indirect damages caused by the use of this document or the use of the company's products, including but not limited to losses caused by loss of business profits, system failure, data or document loss.
- You are aware of the open nature of the Internet. When you connect your product to the Internet, there may be risks such as network attacks, hacker attacks, and virus infections. The company is not responsible for product malfunctions, information leakage, and other problems caused by this, but the company will provide you with product-related technical support in a timely manner.
- When using this product, please strictly follow applicable laws and regulations to avoid infringing third-party rights, including but not limited to intellectual property rights, data rights or other privacy rights. You may not use this product for weapons of mass destruction, biological and chemical weapons, nuclear explosions or any unsafe use of nuclear energy or violation of human rights. • If the content of this document conflicts with applicable laws, the legal provisions shall prevail.

ISUPSDK (General) Development Guide

Table of contents

Chapter 1 Overview	1
1.1 Introduction.....	1
1.2 System requirements.....	1
1.3 Functions and components.....	1
1.3.1 Functional modules.....	1
1.3.2 Service Components.....	2
1.3.3 Relationship between functions and services.....	4
1.4 Update History.....	5
1.5 Notes.....	9
Chapter 2 Device Registration	10
2.1 Device Registration Message Processing Flow.....	10
Chapter 3 Preview	14
Chapter 4 File Search	23
Chapter 5 Playback	27
Chapter 6 Voice Intercom	38
6.1 Enable Voice Intercom.....	38
6.2 Voice forwarding.....	42
6.3 Enabling voice intercom via streaming media service.....	44
Chapter 7 PTZ Control	53
Chapter 8 Alarm or Event Configuration	54
Chapter 9 Enable monitoring service to receive alarms	58
Chapter 10 Equipment Maintenance	63
Chapter 11 Storage Management	64
11.1 Enabling storage services.....	65
11.2 Uploading files to storage services.....	70

11.3 Downloading Files from Storage Services.....	73
Chapter 12 Parameter Configuration	78
12.1 Configuring Power Modes.....	80
Chapter 13 Integration via the Transfer Text Protocol	83
Chapter 14 Interface Reference	85
14.1 Registration Module.....	85
14.1.1 Basic Interface.....	85
14.1.2 Configuration and Control Interface.....	93
14.1.3 Monitoring Interface.....	111
14.1.4 Preview interface.....	111
14.1.5 Playback interface.....	114
14.1.6 Voice Intercom Interface.....	123
14.1.7 Message transmission interface.....	127
14.2 Alarm Monitoring Module.....	136
14.2.1 Basic Interface.....	136
14.2.2 Configuration and control interface.....	141
14.2.3 Monitoring Interface.....	143
14.3 Flow acquisition module.....	144
14.3.1 Basic interface.....	144
14.3.2 Preview Interface.....	150
14.3.3 Playback Interface.....	152
14.3.4 Voice Intercom Interface.....	155
14.4 Storage Management Module.....	158
14.4.1 Basic Interface.....	158
14.4.2 Monitoring Interface.....	164
14.4.3 Client-related interfaces.....	165

Chapter 15 Callback Functions	171
15.1 ASYNC_RESPONSE_CB	171
15.2 DEVICE_REGISTER_CB	171
15.3 ECMSCallback	173
15.4 EHomeMsgCallBack	174
15.5 EHomeSSMsgCallBack	174
15.6 EHomeSSRWCallBack	175
15.7 EHomeSSRWCallBackEx	176
15.8 EHomeSSStorageCallBack	177
15.9 xctinCbc	177
15.10 fVoiceDataCallBack	178
15.11 nCnCbc	179
15.12 PASSTHROUGHDATA CALLBACK	180
15.13 PLAYBACK_DATA_CB	181
15.14 PLAYBACK_NEWRINK_CB	181
15.15 PREVIEW_DATA_CB	182
15.16 PREVIEW_NEWRINK_CB	183
15.17 VOICETALK_DATA_CB	183
15.18 VOICETALK_NEWRINK_CB	184
Appendix A. Structures and Enumerations	330
A.1 Data Structure.....	330
A.1.1 NET_DVR_TYPE_NAME	330
A.1.2 NET_EHOME_ADB_DEBUG	330
A.1.3 NET_EHOME_ALARM_HOST_LOG	330
A.1.4 NET_EHOME_ALARM_HOST_LOG_COND	331
A.1.5 NET_EHOME_ALARM_INFO	332
A.1.6 NET_EHOME_ALARM_ISAPI_INFO	333
A.1.7 NET_EHOME_ALARM_ISAPI_PICDATA	334

A.1.8 NET_EHOME_ALARM_LISTEN_PARAM	334
A.1.9 NET_EHOME_ALARM_MPDCDATA	335
A.1.10 NET_EHOME_ALARM_MSG	336
A.1.11 NET_EHOME_ALARM_STATUS_UNION	338
A.1.12 NET_EHOME_ALARM_TIME_CFG	338
A.1.13 NET_EHOME_ALARM_TIME_COND	339
A.1.14 NET_EHOME_ALARMIN_CFG	339
A.1.15 NET_EHOME_ALARMIN_COND	340
A.1.16 NET_EHOME_ALARMIN_LINKAGE_TYPE	340
A.1.17 NET_EHOME_ALARMOUT_CFG	341
A.1.18 NET_EHOME_ALARMOUT_STATUS_CFG	342
A.1.19 NET_EHOME_ALARMWIRELESSINFO	342
A.1.20 NET_EHOME_AMS_ADDRESS	342
A.1.21 NET_EHOME_ASYNC_RESP_CB_DATA	343
A.1.22 NET_EHOME_BINARY_RECV_DATA	343
A.1.23 NET_EHOME_BINARY_SEND_DATA	346
A.1.24 NET_EHOME_BLACKLIST_SEVER	349
A.1.25 NET_EHOME_CHAN_STATUS_CHANGED	349
A.1.26 NET_EHOME_CHAN_TIMING_STATUS_SINGLE	350
A.1.27 NET_EHOME_CHAR_EFFECT	350
A.1.28 NET_EHOME_CID_INFO	351
A.1.29 NET_EHOME_CID_INFO_INTERNAL_EX	352
A.1.30 NET_EHOME_CID_INFO_PICTUREINFO_EX	353
A.1.31 NET_EHOME_CID_PARAM	353
A.1.32 NET_EHOME_CMS_LISTEN_PARAM	354
A.1.33 NET_EHOME_CMSCB_DATA	355
A.1.34 NET_EHOME_COMPONENT_GROUP_INFO	358
A.1.35 NET_EHOME_COMPRESSION_CFG	358

A.1.36 NET_EHOME_COMPRESSION_COND	360
A.1.37 NET_EHOME_CONFIG	360
A.1.38 NET_EHOME_DEV_DEFAULT_PARAM	361
A.1.39 NET_EHOME_DEV_LOG	362
A.1.40 NET_EHOME_DEV_LOG_COND	363
A.1.41 NET_EHOME_DEV_REG_INFO	364
A.1.42 NET_EHOME_DEV_REG_INFO_V12	365
A.1.43 NET_EHOME_DEV_SESSIONKEY	366
A.1.44 NET_EHOME_DEV_SESSIONKEY_EHOME50	366
A.1.45 NET_EHOME_DEV_STATUS_CHANGED	366
A.1.46 NET_EHOME_DEV_TIMING_STATUS	367
A.1.47 NET_EHOME_DEV_WORK_STATUS	367
A.1.48 NET_EHOME_DEVICE_CFG	368
A.1.49 NET_EHOME_DEVICE_INFO	369
A.1.50 NET_EHOME_DISCOVERY_MODE	371
A.1.51 NET_EHOME_DISPLAY_PARAM	371
A.1.52 NET_EHOME_ERRCODE_RET	372
A.1.53 NET_EHOME_ETHERNET	373
A.1.54 NET_EHOME_FACESNAP_REPORT	373
A.1.55 NET_EHOME_FINDCOND	375
A.1.56 NET_EHOME_FINDDATA	378
A.1.57 NET_EHOME_FLOW_COND	378
A.1.58 NET_EHOME_FLOW_INFO	379
A.1.59 NET_EHOME_GPS_CFG	379
A.1.60 NET_EHOME_GPS_INFO	380
A.1.61 NET_EHOME_HD_STATUS_CHANGED	381
A.1.62 NET_EHOME_HD_TIMING_STATUS_SINGLE	382
A.1.63 NET_EHOME_HEATMAP_REPORT	382

A.1.64 NET_EHOME_HEATMAP_VALUE	383
A.1.65 NET_EHOME_HTTP_PARAM	384
A.1.66 NET_EHOME_HUMAN_FEATURE	384
A.1.67 NET_EHOME_IDENTIFICATION	385
A.1.68 NET_EHOME_IMAGE_CFG	385
A.1.69 NET_EHOME_INSERT_CHARACTER	386
A.1.70 NET_EHOME_INSERT_EFFECT	386
A.1.71 NET_EHOME_INSERT_INFO	387
A.1.72 NET_EHOME_INSERT_INFO_V20	387
A.1.73 NET_EHOME_INSERT_TEXT_INFO	388
A.1.74 NET_EHOME_IPADDR	389
A.1.75 NET_EHOME_IPADDRESS	389
A.1.76 NET_EHOME_IPC_INFO	389
A.1.77 NET_EHOME_LINKAGE_ALARMOUT	390
A.1.78 NET_EHOME_LINKAGE_PTZ	391
A.1.79 NET_EHOME_LISTEN_PREVIEW_CFG	391
A.1.80 NET_EHOME_LISTEN_VOICETALK_CFG	392
A.1.81 NET_EHOME_LOCAL_ACCESS_SECURITY	392
A.1.82 NET_EHOME_LOCAL_DEV_PINGREO	393
A.1.83 NET_EHOME_LOCAL_GENERAL_CFG	393
A.1.84 NET_EHOME_LOCAL_PLAYBACK_PARAM	394
A.1.85 NET_EHOME_LONG_CFG_INPUT	394
A.1.86 NET_EHOME_LONG_CFG_SEND	394
A.1.87 NET_EHOME_MAKE_I_FRAME	395
A.1.88 NET_EHOME_MANUAL_IOOUT_CTRL	395
A.1.89 NET_EHOME_MPDATA	396
A.1.90 NET_EHOME_MPGPS	396
A.1.91 NET_EHOME_NETWORK_CFG	397

A.1.92 NET_EHOME_NEWSLINK_CB_MSG	397
A.1.93 NET_EHOME_NOTICE_PICURL	399
A.1.94 NET_EHOME_NOTIFY_FAIL_INFO	400
A.1.95 NET_EHOME_PASSTHROUGH_PARAM	401
A.1.96 NET_EHOME_PASSWORD_CFG	401
A.1.97 NET_EHOME_PIC_CFG	402
A.1.98 NET_EHOME_PIC_FILE	403
A.1.99 NET_EHOME_PIC_FILE_COND	404
A.1.100 NET_EHOME_PIXEL_ARRAY_SIZE	405
A.1.101 NET_EHOME_PLANCTRLPARAM	406
A.1.102 NET_EHOME_PLAYBACK_DATA_CB_INFO	406
A.1.103 NET_EHOME_PLAYBACK_DATA_CB_PARAM	406
A.1.104 NET_EHOME_PLAYBACK_INFO_IN	407
A.1.105 NET_EHOME_PLAYBACK_INFO_OUT	409
A.1.106 NET_EHOME_PLAYBACK_LISTEN_PARAM	409
A.1.107 NET_EHOME_PLAYBACK_NEWSLINK_CB_INFO	410
A.1.108 NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM	411
A.1.109 NET_EHOME_PLAYER_PARAM	412
A.1.110 NET_EHOME_POINT	412
A.1.111 NET_EHOME_POSITION_INFO	413
A.1.112 NET_EHOME_POST_PUBLISH_XML	413
A.1.113 NET_EHOME_POST_SCHEDULE	414
A.1.114 NET_EHOME_PPPOECFG	414
A.1.115 NET_EHOME_PRESET_PARAM	415
A.1.116 NET_EHOME_PREVIEW_CB_MSG	415
A.1.117 NET_EHOME_PREVIEW_DATA_CB_PARAM	416
A.1.118 NET_EHOME_PREVIEWINFO_IN	416
A.1.119 NET_EHOME_PREVIEWINFO_IN_V11	417

A.1.120 NET_EHOME_PREVIEWINFO_OUT	417
A.1.121 NET_EHOME_PT_PARAM	418
A.1.122 NET_EHOME_PTXML_PARAM	418
A.1.123 NET_EHOME_PTZ_PARAM	419
A.1.124 NET_EHOME_PUBLISH_SERVERADDR	421
A.1.125 NET_EHOME_PUSHPLAYBACK_IN	421
A.1.126 NET_EHOME_PUSHPLAYBACK_OUT	422
A.1.127 NET_EHOME_PUSHSTREAM_IN	422
A.1.128 NET_EHOME_PUSHSTREAM_OUT	422
A.1.129 NET_EHOME_PUSHVOICE_IN	423
A.1.130 NET_EHOME_PUSHVOICE_OUT	423
A.1.131 NET_EHOME_PZIN_PARAM	423
A.1.132 NET_EHOME_QUEUE_DATA	424
A.1.133 NET_EHOME_QUEUE_DATALIST	424
A.1.134 NET_EHOME_QUEUE_INFO	424
A.1.135 NET_EHOME_QUEUE_ITEM_DATALIST	425
A.1.136 NET_EHOME_REC_FILE	425
A.1.137 NET_EHOME_REC_FILE_COND	427
A.1.138 NET_EHOME_RECORD_CHAN	427
A.1.139 NET_EHOME_REGISTER_LISTEN_MODE	428
A.1.140 NET_EHOME_RELATE_IPC_INFO	428
A.1.141 NET_EHOME_RELEASE_SERVER_INFO	429
A.1.142 NET_EHOME_REMOTE_CTRL_PARAM	430
A.1.143 NET_EHOME_REPLACE_MATERIAL	431
A.1.144 NET_EHOME_SCHEDTIME	432
A.1.145 NET_EHOME_SCREEN_SHOT	432
A.1.146 NET_EHOME_SCREEN_SHOT_EX	432
A.1.147 NET_EHOME_SCREEN_SHOT_RET	433

A.1.148 NET_EHOME_SEND_PARAM	433
A.1.149 NET_EHOME_SERVER_INFO	433
A.1.150 NET_EHOME_SERVER_INFO_V50	434
A.1.151 NET_EHOME_SET_REREGISTER_MODE	436
A.1.152 NET_EHOME_SHOT_PIC	436
A.1.153 NET_EHOME_SS_CENTRAL_PARAM	437
A.1.154 NET_EHOME_SS_CLIENT_PARAM	438
A.1.155 NET_EHOME_SS_CLOUD_PARAM	438
A.1.156 NET_EHOME_SS_EX_PARAM	439
A.1.157 NET_EHOME_SS_INIT_PARAM	439
A.1.158 NET_EHOME_SS_KMS_PARAM	440
A.1.159 NET_EHOME_SS_LISTEN_HTTPS_PARAM	440
A.1.160 NET_EHOME_SS_LISTEN_PARAM	441
A.1.161 NET_EHOME_SS_LOCAL_SDK_PATH	442
A.1.162 NET_EHOME_SS_RW_PARAM	443
A.1.163 NET_EHOME_SS_STORAGE_URI	444
A.1.164 NET_EHOME_SS_TOMCAT_MSG	444
A.1.165 NET_EHOME_SS_TOMCAT_PARAM	446
A.1.166 NET_EHOME_SS_VRB_PARAM	447
A.1.167 NET_EHOME_STOPPLAYBACK_PARAM	447
A.1.168 NET_EHOME_STOPSTREAM_PARAM	447
A.1.169 NET_EHOME_STOPVOICETALK_STM_PARAM	448
A.1.170 NET_EHOME_SWITCH_DAILY_PLAN	448
A.1.171 NET_EHOME_SWITCH_PLAN	449
A.1.172 NET_EHOME_SWITCH_PLAN_PARAM	449
A.1.173 NET_EHOME_SWITCH_WEEKLY_PLAN	450
A.1.174 NET_EHOME_SWITCHE_DAY_OF_WEEK_PLAN	450
A.1.175 NET_EHOME_SYSCOMPONENT_REDUCED_ADDR	450

A.1.176 NET_EHOME_TERM_DEFAULT_GATEWAY	451
A.1.177 NET_EHOME_TERM_IP_ADDRESS	451
A.1.178 NET_EHOME_TERM_TIME_ZONE	452
A.1.179 NET_EHOME_TERMINAL_CONTROL	452
A.1.180 NET_EHOME_TERMINAL_CONTROL_V20	452
A.1.181 NET_EHOME_TERMINAL_INFO	453
A.1.182 NET_EHOME_TERMINAL_NAME	454
A.1.183 NET_EHOME_TERMINAL_PROGRESS	454
A.1.184 NET_EHOME_TERMINAL_REPORT_INFO	455
A.1.185 NET_EHOME_TERMINAL_STATE	455
A.1.186 NET_EHOME_TERMINAL_UPG_ST_REPORT	456
A.1.187 NET_EHOME_TEXT_MESSAGE	456
A.1.188 NET_EHOME_TIME	457
A.1.189 NET_EHOME_TIME_ADJUST_PARAM	458
A.1.190 NET_EHOME_UPGRADE_CONN_PARAM	458
A.1.191 NET_EHOME_VERSION_INFO	458
A.1.192 NET_EHOME_VOICE_TALK_IN	459
A.1.193 NET_EHOME_VOICE_TALK_OUT	459
A.1.194 NET_EHOME_VOICETALK_DATA	460
A.1.195 NET_EHOME_VOICETALK_DATA_CB_INFO	460
A.1.196 NET_EHOME_VOICETALK_DATA_CB_PARAM	460
A.1.197 NET_EHOME_VOICETALK_NEWSLINK_CB_INFO	461
A.1.198 NET_EHOME_VOICETALK_PARA	462
A.1.199 NET_EHOME_VOLUME_DAILY_PLAN	463
A.1.200 NET_EHOME_VOLUME_DAY_OF_WEEK_PLAN	463
A.1.201 NET_EHOME_VOLUME_PLAN	464
A.1.202 NET_EHOME_VOLUME_PLAN_PARAM	464
A.1.203 NET_EHOME_VOLUME_WEEKLY_PLAN	465

A.1.204 NET_EHOME_WEATHER_INFO	465
A.1.205 NET_EHOME_WIRELESS_INFO_CFG	466
A.1.206 NET_EHOME_XML_CFG	466
A.1.207 NET_EHOME_XML_REMOTE_CTRL_PARAM	467
A.1.208 NET_EHOME_ZONE	468
A.2 Enumeration Definition.....	468
A.2.1 EN_ALARM_TYPE	468
A.2.2 LONG_LINK_MSG	474
A.2.3 NET_CMS_ENUM_PROXY_TYPE	474
A.2.4 NET_EHOME_ADDRESS_TYPE	475
A.2.5 NET_EHOME_ADDRFORMAT_CMD	475
A.2.6 NET_EHOME_CALLBACK_TYPE	476
A.2.7 NET_EHOME_CMS_INIT_CFG_TYPE	477
A.2.8 NET_EHOME_CONTROL_TYPE	477
A.2.9 NET_EHOME_DAY_OF_WEEK1	479
A.2.10 NET_EHOME_EALARM_INIT_CFG_TYPE	479
A.2.11 NET_EHOME_ERROR_CODE	480
A.2.12 NET_EHOME_ESTREAM_INIT_CFG_TYPE	481
A.2.13 NET_EHOME_INSERT_STATE	482
A.2.14 NET_EHOME_IPC_TYPE	482
A.2.15 NET_EHOME_LOCAL_CFG_TYPE	483
A.2.16 NET_EHOME_PLANCTRL_TYPE	484
A.2.17 NET_EHOME_PLAY_SCHEDULE_TYPE1	484
A.2.18 NET_EHOME_PLAYBACK_OPERATE_MODE	485
A.2.19 NET_EHOME_POS_MODE	486
A.2.20 NET_EHOME_POWER_ONOFF	486
A.2.21 NET_EHOME_PROGRAM_INSERT_TYPE	487
A.2.22 NET_EHOME_REFRESH_TYPE	487

A.2.23 NET_EHOME_REGISTER_TYPE	487
A.2.24 NET_EHOME_SS_CLIENT_TYPE	489
A.2.25 NET_EHOME_SS_INIT_CFG_TYPE	490
A.2.26 NET_EHOME_SS_MSG_TYPE	490
A.2.27 NET_EHOME_SS_TYPE	491
A.2.28 NET_EHOME_STREAM_TYPE	492
A.2.29 NET_EHOME_TALK_ENCODING_TYPE	492
A.2.30 NET_EHOME_TERMINAL_PLAY_STATE	493
A.2.31 NET_EHOME_TERMINAL_TYPE	494
A.2.32 NET_EHOME_THIRD_PARTY_DATA_TYPE	495
A.2.33 NET_EHOME_TIME_ZONE	495
A.2.34 NET_EHOME_TRANSMINT_PROTOCOL	496
A.2.35 SEARCH_GET_NEXT_STATUS_ENUM	497
A.2.36 SEARCH_TYPE	498
Chapter 16 Commands and Messages	185
16.1 Control Command Message.....	185
16.1.1 ADJUSTTIME	185
16.1.2 GETDEVICEWORKSTATUS	185
16.1.3 PZIN	187
16.1.4 REBOOT	188
16.1.5 RESISTANT GRIND	188
16.1.6 SETDEFENSESTATUS	189
16.1.7 UPDATE	190
16.2 Configuration Command Messages.....	191
16.2.1 GetAlarmDeploymentTime	191
16.2.2 GetAlarmLinkageType	192
16.2.3 GetAlarmTriggerCapture	193
16.2.4 GetAlarmTriggerRecord	194

16.2.5 GCbrtinr	195
16.2.6 GetDevAbility	196
16.2.7 Gxctinr	197
16.2.8 GetHideAlarmArea	198
16.2.9 GetHideAlarmPara	198
16.2.10 GetHideArea	199
16.2.11 GetHidePara	200
16.2.12 GMtinr	201
16.2.13 GMtinr	202
16.2.14 GetPassengerLinePara	203
16.2.15 GetPassengerOSDPara	204
16.2.16 GetPassengerPara	205
16.2.17 GetRecordPlanPara	206
16.2.18 GetRS232Para	207
16.2.19 GetRS485Para	208
16.2.20 GetVILostPara	209
16.2.21 SetAlarmDeploymentTime	209
16.2.22 SetAlarmLinkageType	210
16.2.23 SetAlarmTriggerCapture	212
16.2.24 SetAlarmTriggerRecord	213
16.2.25 Cbrtinr	213
16.2.26 xctinr	214
16.2.27 SetHideAlarmArea	215
16.2.28 SetHideAlarmPara	216
16.2.29 SetHideArea	217
16.2.30 SetHidePara	218
16.2.31 Mtinr	219
16.2.32 Mtinr	220

16.2.33 SetPassengerLinePara	220
16.2.34 SetPassengerOSDPara	221
16.2.35 SetPassengerPara	222
16.2.36 SetRecordPlanPara	223
16.2.37 SetRS232To	224
16.2.38 SetRS485Para	225
16.2.39 SetVILostPara	226
16.3 Request URI	227
16.3.1 ymcbtI	227
16.3.2 ymcnmtinMcbtirmn	227
16.3.3 ymcnmtinMrmn	228
16.4 XML or JSON Messages.....	228
16.4.1 NCCnmtinM	228
16.4.2 NCnmtinM	229
16.4.3 JSON_DasInfo	230
16.4.4 NvnNtictinrBry	230
16.4.5 NvnNtictinrFcCrM	231
16.4.6 NvnNtictinrFcMnM	239
16.4.7 NvnNtictinrFccCmrnM	244
16.4.8 NvnNtictinrMM	256
16.4.9 NvnNtictinrmxrctin	257
16.4.10 NvnNtictinrQnrmM	276
16.4.11 NvnNtictinrnrrrVnrM	278
16.4.12 NvnNtictinrWtinmctinM	281
16.4.13 JSON_ResponseStatus	284
16.4.14 XML_DeviceCap	285
16.4.15 XMvnNtictinrrmnrM	298
16.4.16 XMvnNtictinrNRM	298

16.4.17 XMvnNtictinxctin	305
16.4.18 XMvnNtictinxrcctinrmM	305
16.4.19 XMvnNtictinxFcctinrmM	306
16.4.20 XMvnNtictinxFMvnctinM	307
16.4.21 XMvnNtictinxrrrrrmM	308
16.4.22 XMvnNtictinxrnrmM	308
16.4.23 XMvnNtictinxrctin	309
16.4.24 XMvnNtictinxrnctin	310
16.4.25 XMvnNtictinxrnctinrmM	312
16.4.26 XMvnNtictinxMtinctinM	314
16.4.27 XMvnNtictinxrbcRmvctinM	315
16.4.28 XMvnNtictinxrnctinrmM	316
16.4.29 XMvnNtictinxCntinvnM	318
16.4.30 XMvnNtictinxGrnctinrmM	318
16.4.31 XMvnNtictinxRrmM	320
16.4.32 XMvnNtictinxrrnnrnc	321
16.4.33 XMvnNtictinxrrnxtin	323
16.4.34 XMvnNtictinxcnCnctin	324
16.4.35 XMvnNtictinxrmrrrmM	325
16.4.36 XMvnNtictinxnnBctinrmM	326
16.4.37 XMvnNtictinxVrmM	327
16.4.38 XMvnNtictinxVmrrnmM	328
16.4.39 XML_Status	329
Appendix B. Appendix	499
B.1 Event Types and Details.....	511
B.2 ISUPSDK Error Codes.....	501
B.3 ISAPI Return Codes.....	511
B.4 Functional Modular Error Codes.....	558

B.5 Device Log Types.....	613
B.5.1 Alarm host log type.....	600
B.5.2 Log parameter configuration.....	610
B.6 Supported Network Cameras.....	611
B.7 Supported Network Video Recorders.....	619
B.8 Country and Region Codes.....	622
B.9 Area Code.....	633

Chapter 1 Overview

1.1 Introduction

ISUPSDK is a software development kit based on ISUP (Intelligent Security Uplink Protocol) encapsulation, which can provide third-party platforms with interfaces, commands, library files, etc. for accessing Hikvision products. Supported Hikvision products include network cameras, network dome cameras, digital video recorders (DVRs), network video recorders (NVRs), mobile video recorders, mobile devices, alarm devices, access control devices, decoders, etc. Therefore, third-party platforms can use this SDK to implement functions such as preview, playback, voice intercom, alarm, access control, PTZ control, etc.

1.2 System Requirements

Both Windows and Linux operating systems are supported.

For 32-bit **Windows** operating systems

Windows XP/7/8/Vista and Windows Server 2012/2008/2003

For 64-bit **Windows** operating systems

Windows 7/8/10/ and Windows Server 2012/2008

For 32-bit **Linux** OS

GCC version 4.1.2 and above

CentOS version 5.4 and above, Redhat AS version 5.4 and above, Ubuntu version 9.10 and above, Fedora version 12 and above

For 64-bit **Linux** OS

GCC version 4.1.2 and above

CentOS version 5.4 and above, Redhat AS version 5.4 and above, SUSE version 10 and above

1.3 Functions and Components

1.3.1 Functional Modules

ISUPSDK includes the following functional modules: preview, playback, file search, PTZ control, voice intercom, voice forwarding, alarm or event configuration and reception, device maintenance, storage management and parameter configuration.



Figure 1-1 ISUPSDK functional modules

1.3.2 Service Components

ISUPSDK consists of four service components, namely the Central Management Service (abbreviated as CMS), Stream Media Service (SMS), Alarm Management Service (AMS) and Storage Service (SS).

Service Description

Center Management Service (CMS)

ISUPSDK (General) Development Guide

After the device actively registers with CMS, it can start previewing, file searching, playback, parameter setting, voice intercom, etc.

Streaming service (SMS)

The monitoring service for preview and playback can be enabled. In addition, the SMS IP address must be sent to the device via CMS.

Alarm Management Service (AMS)

Enable alarm monitoring service and receive alarm information (such as motion detection alarm, video loss alarm, video occlusion alarm, abnormal behavior detection alarm, passenger flow statistics alarm, heat map alarm, image recognition snapshot alarm, GPS alarm, etc.). In addition, the IP address of AMS must be sent to the device through CMS.

Storage Service (SS)

Can save, upload and download pictures.



The SDK does not provide interaction and communication solutions between services, which need to be implemented by yourself.

Component Description

Table 1-1 Windows operating system

Module Name	Included Files
Public header files	HCISUPPublic.h
Register Module	HCISUPCMS.dll
	HCISUPCMS.lib
	HCISUPCMS.h
Flow acquisition module	HCISUPStream.dll
	HCISUPStream.lib
	HCISUPStream.h
Alarm monitoring module	HCISUPAlarm.dll
	HCISUPAlarm.lib
	HCISUPAlarm.h
Storage Management Module	HCISUPSS.dll

ISUPSDK (General) Development Guide

Module Name	Included Files
	HCISUPSS.lib
	HCISUPSS.h

Table 1-2 Linux operating system

Module Name	Included Files
Public header files	HCISUPPublic.h
Register Module	libHCISUPCMS.so
	HCISUPCMS.h
Flow acquisition module	libHCISUPStream.so
	HCISUPStream.h
Alarm monitoring module	libHCISUPAlarm.so
	HCISUPAlarm.h

1.3.3 Relationship between functions and services

The implementation of ISUPSDK functional modules depends on service components or the cooperation between service components.

Preview and Playback

Relies on the cooperation of the Central Management Service (CMS) and the Streaming Media Service (SMS).

File search, PTZ control, voice forwarding, device maintenance, alarm or event configuration and parameter configuration

Depends on Central Management Service (CMS).

Voice intercom

Reliance on CMS only or on the cooperation of CMS and SMS.

Alarm or event reception

Relies on the collaboration of Central Management Service (CMS) and Alarm Management Service (AMS).

Storage

management relies on storage services (SS).

1.4 Update History

Version V2.5 Update Summary_2021/09 Fixed the abnormal problem caused by constructing messages.

Version V2.5 Update Summary_2020/10

1. Extended initialization configuration type enumeration *NET_EHOME_SS_INIT_CFG_TYPE*:

Added *NET_EHOME_SS_INIT_CFG_SQLITE3_PATH* (set path to sqlite3 library).

2. Extended registration callback function *DEVICE_REGISTER_CB*:

Added 6 callback data types: *ENUM_DEV_SESSIONKEY_REQ* (SessionKey request information for ISUP version 5.0 devices), *ENUM_DEV_DAS_REREREGISTER* (device re-registration information), *ENUM_DEV_DAS_PINGREO* (device registration heartbeat), *ENUM_DEV_DAS_EHOMEKEY_ERROR* (password verification failure information), *ENUM_DEV_SESSIONKEY_ERROR* (Sessionkey interaction exception) and *ENUM_DEV_SLEEP* (device enters sleep state). 3. Extend the device information

structure *NET_EHOME_DEVICE_INFO*:

Added a new member **byStartDTalkChan** (starting digital talk channel number), which occupies 1 byte.

4. Extended the transmission parameter structure *NET_EHOME_PTXML_PARAM*:

Added a new member *dwHandle* (message handle), which occupies 8 bytes.

5. Added the transmission HTTP page interface *NCMCn*. 6. Added an interface

for setting the heartbeat timeout of ISUPSDK for a 5.0 protocol device *NET_ECMS_SetAliveTimeout*. 7.

Extended the CID information structure

NET_EHOME_CID_INFO:

Added new members **pPicInfoEx** (additional pointer information) and *byRes2* (reserved), occupying 8 bytes.

8. Added additional picture information structure *NET_EHOME_CID_INFO_PICTUREINFO_EX*. 9. Extended

CID alarm extended information structure *NET_EHOME_CID_INFO_INTERNAL_EX*:

Added member **byVideoType** (video type), occupying 16 bytes. 10. Extended

read and write callback function structure *NET_EHOME_SS_RW_PARAM*: Added members

byUseRetIndex (whether to use the index set by the upper layer), *byRes1* (reserved), *pRetIndex* (index set by the upper layer), occupying 7 bytes.

11. Extended device registration information structure *NET_EHOME_DEV_REG_INFO*:

The extended member *bySupport* (functions supported by the device) occupies 1 byte.

12. Extended server information structure *NET_EHOME_SERVER_INFO_V50*: Extended

member *dwCloudPoolId* (cloud storage pool ID), occupying 4 bytes. 13. Extended file

upload client type enumeration *NET_EHOME_SS_CLIENT_TYPE*:

Added *NET_EHOME_SS_CLIENT_TYPE_CENTRAL* (central storage streaming storage protocol).

14. The extended structure *NET_EHOME_SS_EX_PARAM* returned by the extended callback function :

Added new consortium member NET_EHOME_SS_CENTRAL_PARAM (central storage consortium).

15. Added central storage complex NET_EHOME_SS_CENTRAL_PARAM.

Version V2.4 Update Summary_2020/04

1. Added the preview start interface NET_ECMS_StartGetRealStream. 2. Added the preview

request input parameter structure NET_EHOME_PREVIEWINFO_IN. 3. Added the start file search interface

NET_ECMS_StartFindFile. 4. Added the preview stop request parameter structure

NET_EHOME_FINDCOND. 5. Added the interface NET_ECMS_FindNextFile for getting the retrieved

files one by one. 6. Added the video file information query structure NET_EHOME_FINDDATA. 7. Added

the initialization parameter setting interface NET_ESS_Init_V11. 8. Added the initialization parameter

structure NET_EHOME_SS_INIT_PARAM. 9. Extended the output parameter

structure NET_EHOME_VOICE_TALK_OUT for voice intercom request : Added 1 parameter IHandle

(output parameter handle), which occupies 4 bytes. 10. Extended voice intercom request input parameter structure

NET_EHOME_VOICE_TALK_IN:

Added a new parameter **byEncodingType** (voice intercom encoding type), which occupies 9 bytes.

11. Added the voice intercom encoding type enumeration NET_EHOME_TALK_ENCODING_TYPE. 12. Extended the

device registration information structure NET_EHOME_DEV_REG_INFO:

Added 1 new parameter **byMarketType** (registration type), occupying 1 byte.

13. Extended callback type enumeration NET_EHOME_CALLBACK_TYPE : Added 2 new

types: NET_EHOME_PASSTHROUGH_CB (ISAPI passthrough asynchronous callback) and NET_EHOME_V2_VOICETALK_CB

(ISUP2.0 protocol voice intercom asynchronous callback). 14. Extended center management server monitoring

parameter structure NET_EHOME_CMS_LISTEN_PARAM:

Added 2 new parameters: **dwKeepAliveSec** (heartbeat interval) and **dwTimeOutCount** (heartbeat timeout count), which occupy 8 bytes in total. 15.

Extended

registration callback type enumeration NET_EHOME_REGISTER_TYPE:

Added 2 new types: ENUM_DEV_DAS_PINGREO (device registration heartbeat) and

ENUM_DEV_SESSIONKEY_ERROR (Sessionkey interaction exception). 16. Extended human

attribute information structure NET_EHOME_HUMAN_FEATURE : Added 1 parameter byMask (whether to

wear a mask), occupying 1 byte. 17. Extended alarm image data structure uploaded via HTTP

NET_EHOME_ALARM_ISAPI_PICDATA : Added 1 parameter szFilename (path to save image file), occupying 8256 bytes. 18. Modified the settings to enable

the central management server (CMS) log function interface NET_ECMS_SetLogFile:

Modify the parameter **bAutoDel** description to: whether to automatically delete the log file after a period of time: "TRUE"-Yes (default), "FALSE"-No. When it

is "FALSE", log compression will be enabled. Every time 50 log files are generated, log compression will be performed (the 50 generated log files will be

compressed into one compressed file). 19. Modify the settings to enable the storage server (SS) log function

interface NET_ESS_SetLogFile:

ISUPSDK (General) Development Guide

Modify the parameter **bAutoDel** to describe: whether to automatically delete the log file after a period of time: "TRUE"-Yes (default), "FALSE"-No. When it is "FALSE", log compression will be started, and log compression will be performed once every 50 log files are generated (compressing the 50 generated log files into one compressed file).

20. Modify the setting parameters to enable the Alarm Management Server (AMS) log function interface *NET_EALARM_SetLogFile* : Modify the parameter **bAutoDel** description to: whether to automatically delete the log file after a period of time: "TRUE"-Yes (default), "FALSE"-No. When it is "FALSE", log compression will be enabled. Every time 50 log files are generated, log compression will be performed (the 50 generated log files will be compressed into one compressed file). 21. Modify the settings to enable the streaming media server (SMS) log function interface *NET_ESTREAM_SetLogFile* :

Modify the parameter **bAutoDel** to describe: whether to automatically delete the log file after a period of time: "TRUE"-Yes (default), "FALSE"-No. When it is "FALSE", log compression will be started, and log compression will be performed once every 50 log files are generated (compressing the 50 generated log files into one compressed file).

22. Extended the interface *NET_ECMS_SetSDKLocalCfg* for setting the local configuration parameters of the central management server (CMS) and the interface *NET_ECMS_GetSDKLocalCfg* for obtaining the local configuration parameters of the central management server (CMS) : Two new local configuration parameter types are added: **DEV_DAS_PINGREO_CALLBACK** (whether to perform **ENUM_DEV_DAS_PINGREO** callback in the registration callback) and **REGISTER_LISTEN_MODE** (registration listening mode of the CMS module).

23. Added the heartbeat callback setting structure *NET_EHOME_LOCAL_DEV_PINGREO* . 24. Added the registration listening mode structure *NET_EHOME_REGISTER_LISTEN_MODE* . 25. Extended the local configuration type enumeration *NET_EHOME_LOCAL_CFG_TYPE* :

Added 3 new types: **DEV_DAS_PINGREO_CALLBACK** (device heartbeat registration callback), **REGISTER_LISTEN_MODE** (registration listening mode), **STREAM_PLAYBACK_PARAM** (playback local parameter configuration).

26. Extend the local configuration parameter interface *NET_ESTREAM_SetSDKLocalCfg* of the streaming media server (SMS) : Two new local configuration parameter types are added: **COM_PATH** (to obtain the path of HCAapSDKCom (the directory of HCISUP-related dependency libraries) of the SMS module) and **STREAM_PLAYBACK_PARAM** (to set whether to enable playback synchronous reception).

27. Added the status type enumeration *SEARCH_GET_NEXT_STATUS_ENUM* . 28. Extended the terminal log query result structure *NET_EHOME_DEV_LOG* : Added 2 new parameters: **bym** (the difference in hours between the log time and the Universal Standard Time (UTC)) and **bymM** (the difference in minutes between the log time and the Universal Standard Time (UTC)), which occupy 2 bytes in total. 29. Added the interface *NET_ESS_GenerateStorageUri* for generating URL information of storage resources (images, etc.) . 30. Added the URL information structure *NET_EHOME_SS_STORAGE_URI* for storage resources . 31. Added the storage service type enumeration *NET_EHOME_SS_TYPE* . 32. Added the interface *NCnB* for clients to upload resources (images) in the form of memory data . 33. Added the read-write extension callback function structure *NET_EHOME_SS_RW_PARAM* . 34. Added the extension structure *NET_EHOME_SS_EX_PARAM* returned by the callback function . 35. Added the cloud storage structure *NET_EHOME_SS_CLOUD_PARAM* . 36. Added TOMCAT storage information structure *NET_EHOME_SS_TOMCAT_PARAM* . 37. Added KMS storage information structure *NET_EHOME_SS_KMS_PARAM* . 38. Added VRB storage information structure *NET_EHOME_SS_VRB_PARAM* .

Version V2.3.5.1 Update Summary_2020/04

Related products: Network camera with software version 5.5.110 (G4 Wild Protection), device model: DS-2XS6C22F-I(W)(GLG)

1. Extended passive infrared alarm message MnNcnRmM :

Added a new **node <CaptureList> (capture list)**.

2. Extended device capability message XML_DeviceCap (related URI: ymcb):

Added 1 new node <By> (whether to support battery status reporting).

Added 2 **nodes to node <SysCap>: <CnmnM> (whether power consumption mode switching is supported)** and <MnwCnmn> (whether manual power consumption control is supported).

3. Extended device power consumption mode capability message NCCnmnM (related URI: ymcnmnMcbmn ý)

Added 1 new power mode "sleepOrWakeUp" to **devWorkMode**;

Added a new **node "sleepOrWakeUpPrompt" (device sleep or wakeup mode prompt)**.

4. Extended device power consumption mode parameter message NCnmnM (related URI: ymcnmnMmn ý)

Added a new power mode "sleepOrWakeUp" to **devWorkMode**.

5. Added battery status event type (bry) and details NnNcnBy to

Event type and details

6. Extended registration callback type enumeration NET_EHOME_REGISTER_TYPE :

Added 1 new type: ENUM_DEV_SLEEP (device enters sleep state).

7. Added the interface NET_ECMS_WakeUp to wake up the device from sleep mode . For Configuring Power Modes details, see 8. Added 2 error codes to ISUPSDK 8301-“NET_ERR_DEV_SLEEP” (the device is currently in sleep mode).

state), 8302-“NET_ERR_DEV_GOINGTOSLEEP” (about to enter sleep state, estimated 13 seconds).

Version V2.3.1.2 Update Summary_2019/11

Related products: I series network hard disk recorders with software version 4.21.005 (for specific product models, see Supported network hardware

VCR

Added temperature alarm (see MnNcnmmM for details), hard disk error alarm

(For details, see MnNcnmM) and illegal login alarm (For details, see

MnNcnmmM ý to

Event type and details

Version V2.3.1.2 Update Summary_2019/11

Related products: 50 series and 70 series box cameras with software version 5.6.0, 51 series and 71 series indoor domes, 55 series and 75 series outdoor dome cameras, 5A series and 7A series barrel cameras, 7D series dome cameras, 7T series barrel cameras, 72 series bolt cameras, hood integrated machine, 78 series barrel machine, explosion-proof face camera (for specific

Supported IP cameras

product models, see 1. Added command message to realize PTZ PTZ control

control function, see 2. Added command message to set alarm or event alarm or event configuration

parameters, see 3. Added multiple types of network camera alarms or events and their details, see 4. Event Type and Details

Added device maintenance related command messages, such as upgrade, obtain device information, time calibration, restart, Equipment Maintenance

ISUPSDK (General) Development Guide

5. Added command message configuration parameters, such as network, encoding, image, OSD, video, serial port and other parameters, see [Parameter configuration](#)
6. Modify the exception callback function registration operation during preview and playback to a mandatory operation, see and [7. Parameter configuration](#) [Preview](#) [Playback](#)
following 4 error codes to `ISUPSDK : 3601`-"`NET_SS_CLIENT_ERR_CMS_TOKEN_FAIL`",
`3602`-"`NET_SS_CLIENT_ERR_KMS_UPLOAD_FAIL`"`3603`-
`"NET_SS_CLIENT_ERR_CLOUD_POOLIST_FAIL"` `3604`-
`"NET_SS_CLIENT_ERR_CLOUD_BESTNODE_FAIL"`.
8. Added storage callback function (`EHomeSSStorageCallBack`) and read/write callback function (`EHomeSSRWCallBack`)
For details on the constraints between them, see the structure `NET_EHOME_SS_LISTEN_PARAM`.

Version V2.3 Update Summary_2019/08

Create a new document.

1.5 Notes

- Under Windows 32-bit or Linux 32-bit operating systems, the Central Management Service (CMS) can support up to 2048 devices.
Central Management Service (CMS) supports up to 10,000 devices.
- Under Windows 32-bit or Linux 32-bit operating systems, it supports up to 512 channels of preview or playback; under Windows 64-bit or Linux 64-bit operating system, it can support up to 2048 channels for preview or playback.
Supports 512-channel preview or playback, regardless of the operating system.
- The Alarm Management Service (AMS) can receive up to 2,000 alarms per second.
- The storage service (SS) can upload or download up to 20 images per second, and the size of each image cannot exceed 200 KB.

Chapter 2 Device Registration

Before implementing the device function through ISUPSDK, you must first log in to the device through the web page to enable the ISUP function and Register the device on the CMS service.

Prerequisites

- Ensure that `NET_ECMS_Init` has been called to initialize CMS.
- Make sure `NET_ECMS_StartListen` has been called to start the listening service.

Steps



The web configuration interface of different devices is different. The following steps are for reference only. For details, please refer to the device user manual.

1. Open the browser and log in to the device through the web page.
2. Click 3. Configuration → network → Advanced Configuration → Platform access Enter the platform access configuration interface.
Select the platform access method as *ISUP*.
4. Check to **Enable** the ISUP function of the device.
5. Set the ISUP version, server address, port number, device ID and key.



The key must be configured only when ISUP version 5.0 is selected.

6. Register the device.



Device registration messages need to be processed according to a certain logical process, see [Device registration message processing flow](#)

2.1 Device registration message processing flow

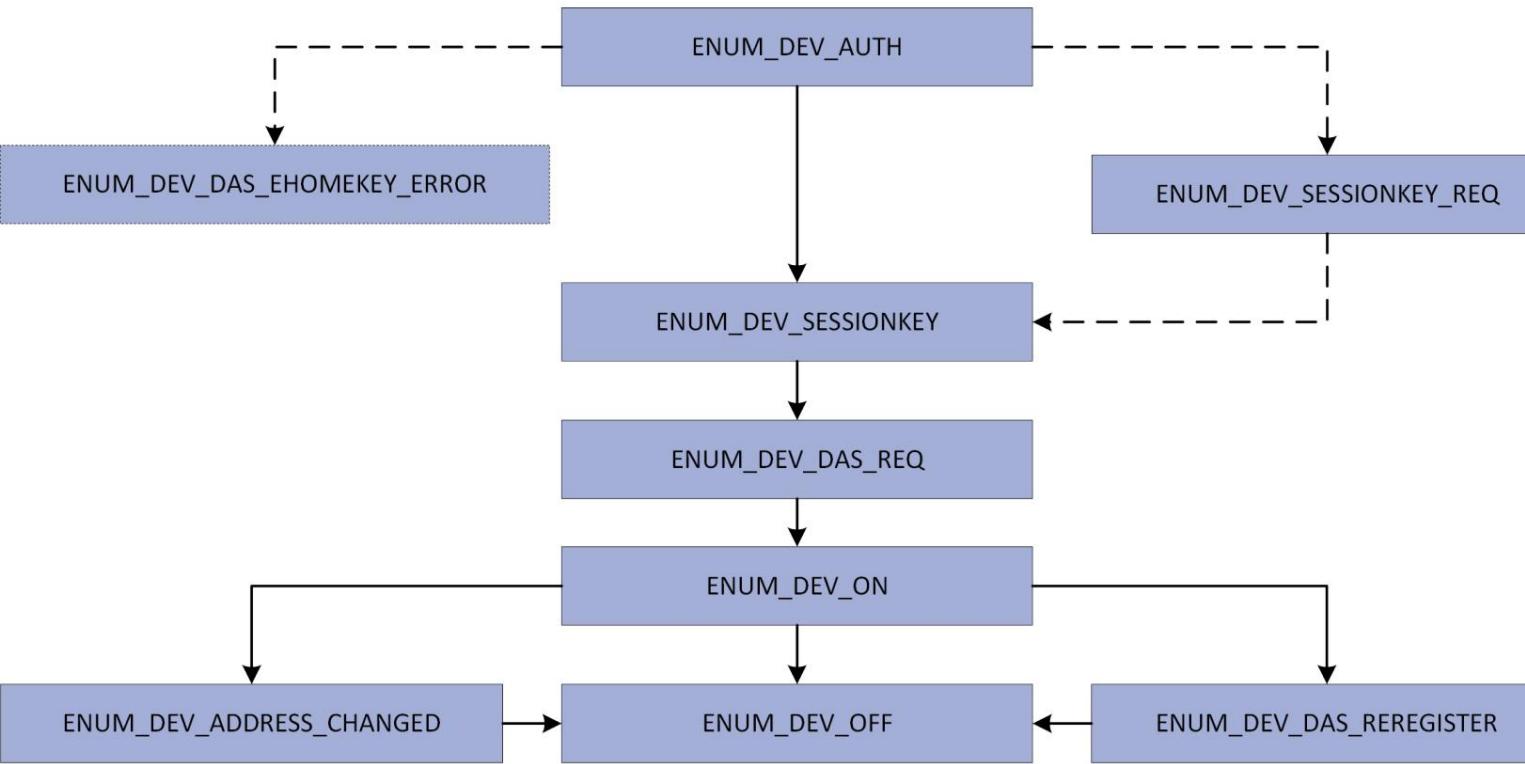


Figure 2-1 Device registration message processing flow chart

Sample

code for the sample device registration message processing flow

```

BOOL _stdcall EHOME_REGISTER(LONG iUserID, DWORD dwDataType, void Br DWORD dwOutLen, void
    nBr DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)

    { //Device online
        //B points to the device registration information structure NET_EHOME_DEV_REG_INFO_V12 //nB
        points to the service information structure NET_EHOME_SERVER_INFO
        NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12 )Br NET_EHOME_SERVER_INFO
        *pServInfo = (NET_EHOME_SERVER_INFO )nBr //Note: The string in the structure is encoded in UTF-8 //
        Output device information rntf(vc Online] DeviceID[%s]\n",
        pDevInfo-
            >struRegInfo.byDeviceID); //Set the IP address of the alarm server pServInfo->dwAlarmServerType
        = 1; //two alarm server types are
        supported: TCP and UDP //Set the address of the TCP alarm server strcpy(pServInfo->struTCPAlarmSever.szIP,
        "10.21.43.15"); pServInfo-
            >struTCPAlarmSever.wPort = 7332; //Set UDP The address of the alarm server
            strcpy (pServInfo->struUDPAlarmSever.szIP, "10.21.43.15");
            pServInfo->struUDPAlarmSever.wPort =
            7332
    }
}
  
```

ISUPSDK (General) Development Guide

```

//Note: Filter devices according to device ID for registration. If the device is not allowed to be registered, it will return FALSE
return TRUE;

} else if (ENUM_DEV_OFF == dwDataType)

{ //Device offline //
Output device
information rntf(vc ffln DeviceID[%s]\n", pDeviceInfo->struRegInfo.byDeviceID);
return TRUE;

} else if (ENUM_DEV_AUTH == dwDataType)
{

//For authentication of devices that support ISUP5.0, EHomeKey must be
configured NET_EHOME_DEV_REG_INFO_V12 *pDeviceInfo = (NET_EHOME_DEV_REG_INFO_V12 )Br //Output device information

rntf(vc ffln DeviceID[%s] IP[%s]\n", pDeviceInfo->struRegInfo.byDeviceID, pDeviceInfo-
>struRegInfo.struDevAdd.szIP); char
szEHomeKey[32] = "i#Rhg&8D5IP";
rcy((cr)nBr szEHomeKey); //Set device EhomeKey for return TRUE; } else if ntictin

(ENUM_DEV_SESSIONKEY == dwDataType)
{

//For authentication of devices that support ISUP5.0, EHomeKey must be configured
NET_EHOME_DEV_REG_INFO_V12 *pDeviceInfo = (NET_EHOME_DEV_REG_INFO_V12 )Br rntf(vc SessionKey] DeviceID[%s]\n",
pDeviceInfo->struRegInfo.byDeviceID); NET_EHOME_DEV_SESSIONKEY struSessionkey = {0};
memcpy(struSessionkey.sDeviceID, pDeviceInfo->struRegInfo.byDeviceID,
MAX_DEVICE_ID_LEN); memcpy(struSessionkey.sSessionKey, pDeviceInfo->struRegInfo.bySessionKey, MAX_MASTER_KEY_LEN);
NET_ECMS_SetDeviceSessionKey(&struSessionkey);

//For devices that support ISUP5.0, the alarm module also needs to be authenticated first, so the device SessionKey must
be available //If AMS and CMS are installed in the same program, please call NET_EALARM_SetDeviceSessionKey directly
NET_EALARM_SetDeviceSessionKey(&struSessionkey); //If AMS and
CMS are not installed in the same program, please transfer SessionKey to AMS first, and then add return TRUE;

} else if (ENUM_DEV_DAS_REQ == dwDataType)

{ //During registration, the device requests DAS information
char szDASIP[128] = "10.21.43.15"; //DAS IP address DWORD dwPort =
6666; //DAS port No. rntf((cr)nBr "{\"Type\":\"DAS\",
\"DasInfo\":{\"Address\":\"%s\",
\"Domain\":\"test.ys7.com\",
\"ServerID\":\"das_%s_%d\", \"Port\":%d,\"UdpPort\":%d}\n", szLocalIP, szLocalIP,
dwPort, dwPort, dwPort);

} else if (ENUM_DEV_ADDRESS_CHANGED == dwDataType)

{ //Device IP address changes (for example, when the device is in a mobile network or across base stations)
NET_EHOME_DEV_REG_INFO_V12 *pDeviceInfo = (NET_EHOME_DEV_REG_INFO_V12 )Br //Output device information

```

ISUPSDK (General) Development Guide

```
rntf(vc IP Changed] DevicelD[%s] New IP[%s]\n", pDevInfo->struRegInfo.byDeviceID, pDevInfo->struRegInfo.struDevAdd.szIP);

} else if (ENUM_DEV_DAS_REREGISTER == dwDataType) { //

When the device information changes, re-register the registered
device NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12 )Br //Output
device information
    rntf(vc ReRegister] DevicelD[%s] IP[%s]\n", pDevInfo->struRegInfo.byDeviceID, pDevInfo->struRegInfo.struDevAdd.szIP);
    return TRUE;

} return TRUE;
}
```

Chapter 3 Preview

Through preview, you can remotely obtain real-time video or audio information of the monitored area, so as to know the status of people, vehicles, objects, etc. The preview function integrated through ISUPSDK requires the central management service (CMS), streaming service (SMS) and playback library.

Prerequisites •

Make sure that *NET_ECMS_Init* and *NET_ESTREAM_Init* have been called to initialize CMS and SMS. • Make sure that *NET_ECMS_StartListen* has been called to start the CMS listening service in order to receive device registration information.

Steps

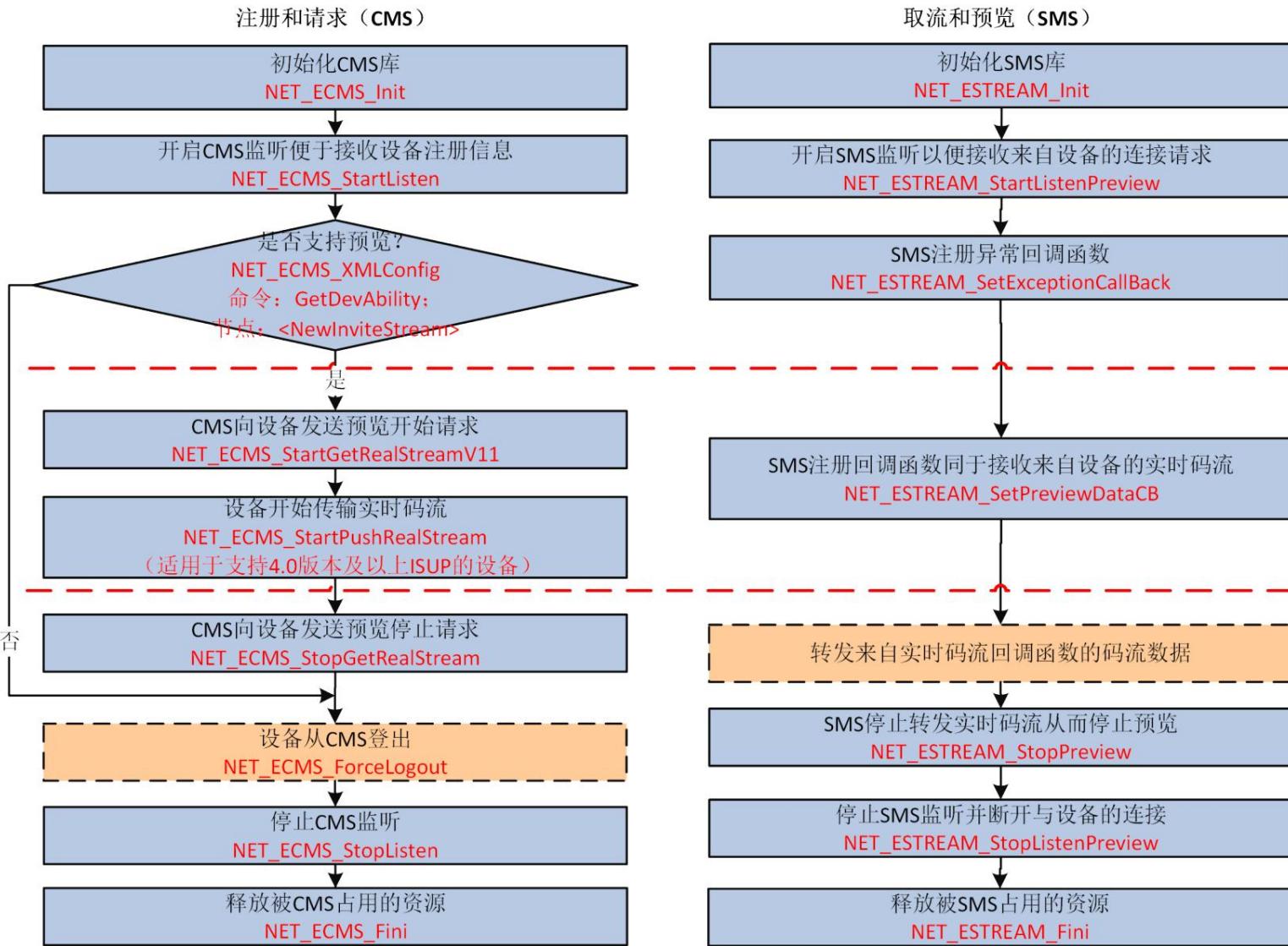


Figure 3-1 Preview interface call flow chart

1. Call `NCMMCn` and command `GetDevAbility` to obtain the device capability and determine whether preview is supported.

The device capability set is returned by `pOutBuf`.

If supported, the node `<NewInviteStream>` will be returned and you can continue with the following steps.

Otherwise, end the task.

2. Call `NET_ESTREAM_StartListenPreview` to start the SMS listening service and receive the connection request from the device. 3. Call `NRMxcnCBc` to register the exception callback function for SMS.

The exception information will be called back to the registered callback function. Please handle the exception in time to avoid blocking.

ISUPSDK (General) Development Guide

4. Call NET_ECMS_StartGetRealStreamV11 to send a preview start request from the CMS to the device.

The address and port number of the SMS are sent to the device, and a session ID is automatically assigned to the CMS.

5. Call NET_ESTREAM_SetPreviewDataCB to register the callback function for SMS to receive the real-time stream from the device. 6. Call NET_ECMS_StartPushRealStream to send the real-time stream transmission request from CMS to the device.

The device automatically connects to SMS and starts sending real-time stream to SMS.

7. Optional operation: Forward the code stream from the SMS real-time stream callback function.



The decoding and display of the preview code stream is implemented by the

client software. 8. Call NET_ESTREAM_StopPreview to stop the real-time stream forwarding of SMS. 9. Call

NET_ECMS_StopGetRealStream to send the preview stop request from CMS to the device.

The device stops transmitting the real-time code stream. 10. Call NET_ESTREAM_StopListenPreview to stop the SMS listening service and disconnect it from the device. 11. Call NET_ECMS_ForceLogout to force the device to log out.

Sample

code for sample preview



In this sample code, CMS and SMS are installed in the same application on the same computer.

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

LONG ILoginID = -1;
LONG ILinkHandle = -1;
LONG IRealHandle = -1;
FILE V      = NULL;

/// ...

BOOL CALLBACK RegisterCallBack(LONG IUserID, DWORD dwDataType, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDeviceInfo = (NET_EHOME_DEV_REG_INFO )pUser

        if (pDeviceInfo != NULL)
        {
            ILoginID = IUserID;
            rntf(nn IUserID: %d, Device ID: %s\n", ILoginID, pDeviceInfo->byDeviceID);
        }
    }
}
```

```

} //Input parameters
NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO )nBr pServerInfo->dwTimeOutCount =
6; //Heartbeat timeout times pServerInfo->dwKeepAliveSec = 15; //Heartbeat
interval

} else if (ENUM_DEV_OFF == dwDataType) { rntf(n

IUserID: %d\n", IUserID);
NET_ECMS_ForceLogout(IUserID); } else if

(ENUM_DEV_AUTH == dwDataType) {

//For device authentication that supports ISUP5.0, the authentication key
NET_EHOME_DEV_REG_INFO_V12 must be set *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12 )Br
//Output device
informationrntf(vc Auth] DeviceID[%s] IP[%s]\n", pDevInfo->struRegInfo.byDeviceID, pDevInfo-
>struRegInfo.struDevAdd.szIP);
char szEHomeKey[32] = "i#Rhg&8D5IP"; //Assume the key is i#Rhg&8D5IP rcy((cr)\nBr
szEHomeKey); //Set device EhomeKey for } else if (ENUM_DEV_SESSIONKEY ntictin

== dwDataType) { //For device authentication that supports

ISUP5.0, EHomeKey must be configured
NET_EHOME_DEV_REG_INFO_V12 *pDevInfo = (NET_EHOME_DEV_REG_INFO_V12 )Br rntf(vc 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;

}

/////////////////////////////// //Process real-time code stream data

BOOL InputStreamData(BYTE byDataType, char* Br int iDataLen) {

(IN      == NULL)
{
IN      = fopen("Test.mp4", "wb"); rntf(v data
to Test.mp4\n");
}

(IN      NULL)
{
wr(Brn1V) //Call back the real-time code stream and save the video file

```

```
}

//Call the playback library to decode and display the code stream
for
preview/* if(1 == byDataType)
{
    if (!PlayM4_GetPort(&m_lPort))
    {
        return FALSE;

    } if (!PlayM4_SetStreamOpenMode(m_lPort, STREAME_REALTIME))
    {
        return FALSE;

    } // Input the first 40 bytes
of the header if(!PlayM4_OpenStream(m_lPort, (unsigned char )Br (DWORD)iDataLen, 2*1024*1024))
{
    return FALSE;

} if(!PlayM4_Play(m_lPort, hWnd))
{
    return FALSE;
}

} else
{
    for (int i=0; i<1000; i++)
    {
        BOOL bRet = PlayM4_InputData(m_lPort,(unsigned char )Br (DWORD)iDataLen); if (!
bRet) {

            if ( i >=999) {

                rntf(yM4n
failed, error code: %d!", PlayM4_GetLastError(m_lPort));
            }
            Sleep(2);
        }
        } else
        {
            break;
        }
    }
}

*/ return TRUE;
}

/// ...
```

```

if (NULL == pPreviewCBMsg)
{
    return ;
}

} IRealHandle = IPreviewHandle;
InputStreamData(pPreviewCBMsg->byDataType, (char*)pPreviewCBMsg->pRecvdata, pPreviewCBMsg->dwDataLen);
}

/// ...

BOOL CALLBACK fnPREVIEW_NEWLINK_CB(LONG IPreviewHandle, NET_EHOME_NEWLINK_CB_MSG
*pNewLinkCBMsg, void *pUserData) {

    ILinkHandle = IPreviewHandle;
    rntf(Cbc of preview listening, Device ID: %s, Channel: %d\n", pNewLinkCBMsg->szDeviceID, pNewLinkCBMsg-
>dwChannelNo);

    //Callback parameters of
    preview data NET_EHOME_PREVIEW_DATA_CB_PARAM struDataCB
    = {0}; struDataCB.fnPreviewDataCB = fnPREVIEW_DATA_CB;
    struDataCB.byStreamFormat = 0; //Encapsulation format: 0-PS format

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

    } rntf(NRMrvwCBn)

    return TRUE;
}

/// ...

NRMxctinCbc(0 0, rmxctinCbc NULL);

void CALLBACK rmxctinCbc(WR dwType, LONG iUserID, LONG iHandle, void* pUser) {

    if(EHOME_PREVIEW_EXCEPTION == dwType)
    {
        //Preview exception
        rntf(rvw xctin handle=%d, Error:%d", iHandle, dwError);

    } else if (EHOME_PLAYBACK_EXCEPTION == dwType)
    {
        //Replay
        exceptionrntf(ybc xctin handle=%d, Error:%d", iHandle, dwError);

    } else if (EHOME_AUDIOTALK_EXCEPTION == dwType)
    {
        //Voice talk (forwarding)
        exceptionrntf(rm VoiceTalk xctin handle=%d, Error:%d", iHandle, dwError);
    }
}

```

```

}

}

void main()
{
    //SMS gets the code stream after the monitoring service is turned on

    // Initialize SMS library
    NET_ESTREAM_Init();

    //Preview listening
    parameters 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
    listening port number struListen.fnNewLinkCB = fnPREVIEW_NEWSLINK_CB; //
    Preview request callback
    function struListen.pUser = NULL; struListen.byLinkMode = 1; //0-TCP, 1-UDP

    //Start monitoring service
    LONG lHandle = NET_ESTREAM_StartListenPreview(&struListen);
    if(lHandle < -1)
    {
        rntf(NRMRnrvw failed, error code: %d\n", NET_ESTREAM_GetLastError());
        NET_ESTREAM_End();
        return
    }

    } rntf(NRMRnrvwn)
    //...
}

// Initialize CMS library
NET_ECMS_Init();

//Registered listening
parameters 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;

//Start monitoring and receive device registration information
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
if(lListen < -1) {

    rntf(NCMRn failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

} rntf(NCMRnn)

```

```
while(1 {

    Sleep(1000); //The following rtin should be done when the rtin is completed if(ILoginID >= 0) {

        break;
    }
}

//Input parameters of preview
request NET_EHOME_PREVIEWINFO_IN_V11 struPreviewIn = {0};
struPreviewIn.iChannel = 1; //Channel number
struPreviewIn.dwLinkMode = 1; //0-TCP, 1-UDP
struPreviewIn.dwStreamType = 0; //Stream type: 0-main stream, 1-sub stream, 2-third stream
memcpy(struPreviewIn.struStreamSever.szIP, "10.16.2.123", sizeof("10.16.2.123")); //SMS IP address
struPreviewIn.struStreamSever.wPort = 8003; //SMS port number, which must be consistent with the listening port number

//Preview the output parameters of the request
NET_EHOME_PREVIEWINFO_OUT struPreviewOut = {0};

//Preview
request if(!NET_ECMS_StartGetRealStreamV11(ILoginID, &struPreviewIn, &struPreviewOut)) {

    rntf(NCMrGRrmV11 failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
} rntf(NCMrGRrmV11n)

//Input parameters of stream
transmission request NET_EHOME_PUSHSTREAM_IN
struPushStreamIn = {0}; struPushStreamIn.dwSize =
sizeof(struPushStreamIn); struPushStreamIn.lSessionID = struPreviewOut.lSessionID; //Session ID of preview request

// Output parameters of code stream transmission request
NET_EHOME_PUSHSTREAM_OUT struPushStreamOut = {0};

//Send a request to the device and start transmitting
real-time stream if(!NET_ECMS_StartPushRealStream(ILoginID, &struPushStreamIn, &struPushStreamOut)) {

    rntf(NCMrRrm failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
} rntf(NCMrRrmn)

Sleep(50000);

//////////////////////////////Sign out
```

```

//Release the resources occupied by CMS
preview request if(!NET_ECMS_StopGetRealStream(lLoginID, struPreviewOut.lSessionID)) {

    rntf(NCMGRrm failed, error code: %d\n", NET_ECMS_GetLastError());
}

//CMS stops listening service
if(!NET_ECMS_StopListen(lListen)) {

    rntf(NCMn failed, error code: %d\n", NET_ECMS_GetLastError());
}

//Release the resources occupied by CMS

NET_ECMS_Fini(); /////////////////

//SMS stops forwarding code
stream if(lRealHandle >=
0) {
    if (!NET_ESTREAM_StopPreview(lRealHandle)) {

        rntf(NRMrvw failed, error code: %d\n", NET_ECMS_GetLastError());
    }
}

//SMS stops monitoring
service if(lHandle
>= 0) {
    if (!NET_ESTREAM_StopListenPreview(lHandle)) {

        rntf(NRMnrvw failed, error code: %d\n", NET_ECMS_GetLastError());
    }
}

} //Release the resources occupied by SMS
NET_ESTREAM_End();

//Release file resources
(V != NULL) {

    c(V)
    IN      = NULL;
}

rntf(xn)
}

```

Subsequent

processing • Call *NET_ECMS_StopListen* to stop the CMS listening service. • Call *NET_ECMS_Fini* and NET_ESTREAM_Fini to release the resources occupied by CMS and SMS.

Chapter 4 File Search

Video or audio files saved in the device storage can be searched for playback through the Central Management Service (CMS).

Prerequisites

- Make sure that `NET_ECMS_Init` has been called to initialize CMS. •

Make sure that `NET_ECMS_StartListen` has been called to start the CMS listening service to receive device registration information.

Steps

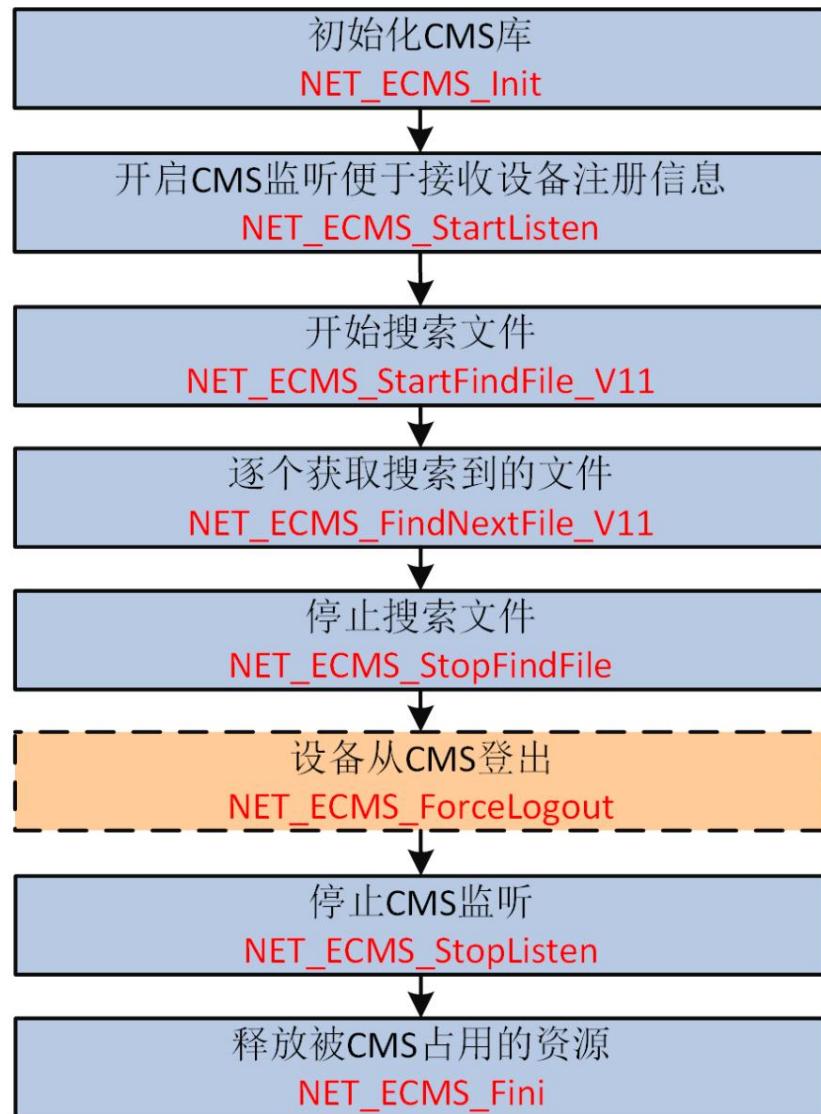


Figure 4-1 Interface call flow chart for file search

1. Call NET_ECMS_StartFindFile_V11 to start searching for files. **2.**

Call NET_ECMS_FindNextFile_V11 to get the searched files one by one. **3.** Call NET_ECMS_StopFindFile to stop searching for files. **4.** Optional operation: Call NET_ECMS_ForceLogout to force the device to log out.

Sample code for sample file lookup

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

LONG ILoginID = -1;
LONG ILinkHandle = -1;
LONG IPlayHandle = -1;
FILE V      = NULL;

// Initialize CMS
NET_ECMS_Init();

//Registered listening
parameters 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;

//Start monitoring service and receive device registration information
LONG IListen = NET_ECMS_StartListen(&struCMSListenPara); if(IList
< -1) {

    rntf(NCMrn failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMrnn)

while(1)
{
    Sleep(1000); //The following operations can only be performed after registration is
    complete if(ILoginID >=
    0) {
        break;
    }
}

//Find video file
NET_EHOME_REC_FILE_COND struFindCond = {0};
struFindCond.dwChannel = 1; //Channel number, starting from 1
```

```

struFindCond.dwRecType = 0x //All types
struFindCond.dwstartIndex = 0; //Search start position
struFindCond.dwMaxFileCountPer = 5; //Maximum number of files that can be searched at one time

//yyyyyy
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;

//yyyyyy
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 ISearchType = 0; //Search for video files
LONG IFileHandle = NET_ECMS_StartFindFile_V11(ILoginID, ISearchType, &struFindCond, sizeof(struFindCond)); //Start file
search if
(IFileHandle < 0) {

    rntf(NCMrFnFV11 failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMrFnFV11n)

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

NET_EHOME_REC_FILE struFileInfo = {0};

//You can search for files in the
thread
while(1) {
    IRet = NET_ECMS_FindNextFile_V11(IFileHandle, &struFileInfo, sizeof(struFileInfo)); //Get search results one by one
    if (IRet == ENUM_GET_NEXT_STATUS_SUCCESS)
    {
        if (struFileInfo.dwFileSize / 1024 == 0) {

            rntf(cmrFnwF)

        } else if (struFileInfo.dwFileSize / 1024 > 0 && struFileInfo.dwFileSize /(1024*1024) == 0) {

            rntf(cmrFnwF1024)
        }
    }
}

```

```

else
{
    rntf(cmMrFnwF /1024/1024);

}

rntf(Fnm %02d:      Filesize[%s], StarTime[%04d-%02d-%02d %02d:%02d:%02d], StopTime[%04d-%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); //yyyyyy

memcpy(szFileName, struFileInfo.sFileName, MAX_FILE_NAME_LEN);

} else
{
    if (lRet == ENUM_GET_NETX_STATUS_NEED_WAIT) {

        Sleep(5);
        ctnin
    }

    if ((lRet == ENUM_GET_NETX_STATUS_NO_FILE) || (lRet == ENUM_GET_NEXT_STATUS_FINISH)) {

        rntf(N more break;      n)

    } else if(lRet == ENUM_GET_NEXT_STATUS_NOT_SUPPORT) {

        rntf(vc does not support\n"); break;

    } else
    {
        rntf(F      to na      for the server is busy or network failure\n");
        break;
    }
}

//CMS stops listening service
if(INET_ECMS_StopListen(lListen)) {

    rntf(NCMn failed, error code: %d\n", NET_ECMS_GetLastError());
}

//Release the resources occupied by CMS
NET_ECMS_Fini();

```

Subsequent

processing • Call NET_ECMS_StopListen to stop the CMS listening service. • Call NET_ECMS_Fini to release the resources occupied by CMS.

Chapter 5 Replay

Playback means playing the stored and searched video files. Playback through ISUPSDK depends on the central management service. (CMS), streaming services (SMS), and playback libraries.

Prerequisites

- Make sure that *NET_ECMS_Init* and *NET_ESTREAM_Init* have been called to initialize CMS and SMS.
- Make sure that *NET_ECMS_StartListen* has been called to start the CMS listening service in order to receive device registration information.

Steps

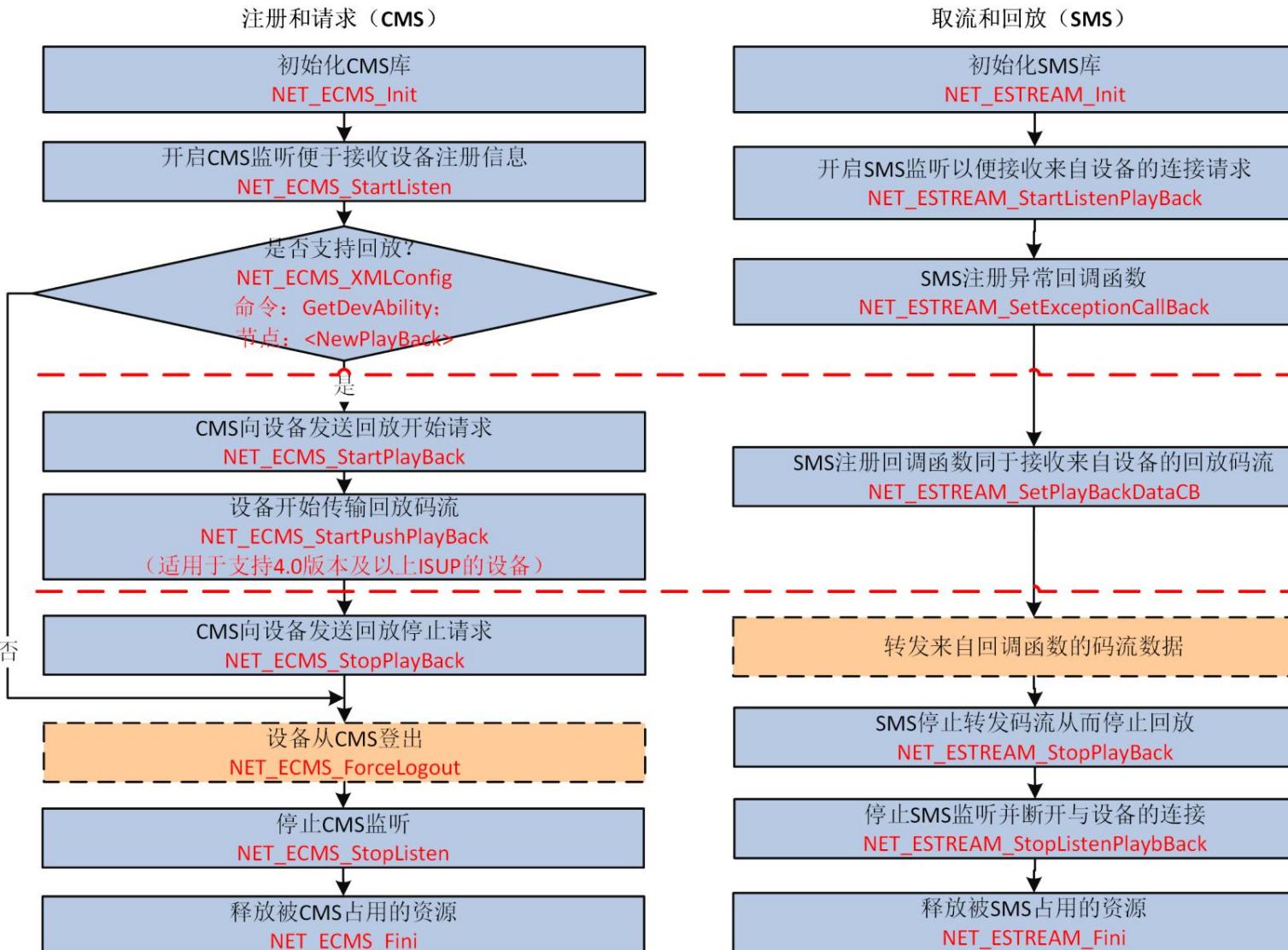


Figure 5-1 Playback interface call flow chart

1. Call NCMMCn and command GetDevAbility to obtain the device capability and determine whether playback is supported.

The device capability set is returned by pOutBuf .

If supported, the node <NewPlayback> will be returned and you can continue with the following steps.

Otherwise, end the task.

2. Call NET_ESTREAM_StartListenPlayBack to start the SMS listening service and receive the connection request from the device.

3. Call NRMxcnCBc to register the exception callback function for SMS.

The exception information will be called back to the registered callback function. Please handle the exception in time to avoid blocking.

ISUPSDK (General) Development Guide

4. Call NET_ECMS_StartPlayBack to send a playback start request from the CMS to the device.

The address and port number of the SMS are sent to the device, and a session ID is automatically assigned to the CMS.

5. Call NET_ESTREAM_SetPlayBackDataCB to register the callback function for SMS to receive the playback code from the device flow.

6. Call NET_ECMS_StartPushPlayBack to send the stream transmission request from CMS to the device. The device will automatically connect to SMS and start sending the stream to SMS. 7. Optional operation: Forward the stream from the SMS stream callback function.



The decoding and display of the playback code stream is implemented by the client software.

8. Call NET_ESTREAM_StopPlayBack to stop SMS stream forwarding. 9. Call NET_ECMS_StopPlayBack to send a playback stop request from CMS to the device.

The device stops transmitting the real-time code stream. 10. Call NET_ESTREAM_StopListenPlayBack to stop the SMS listening service and disconnect it from the device. 11. Optional operation: Call NET_ECMS_ForceLogout to force the device to log out.

Sample

code for sample playback



In this sample code, CMS and SMS are installed in the same application on the same computer.

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

LONG ILoginID = -1;
LONG ILinkHandle = -1;
LONG IPlayHandle = -1;
FILE V      = NULL;

/// ...

BOOL CALLBACK RegisterCallBack(LONG IUserID, DWORD dwDataType, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDevInfo = (NET_EHOME_DEV_REG_INFO )pUser
        if (pDevInfo != NULL)
        {
            ILoginID = IUserID;
        }
    }
}
```

```

rntf(nn lUserID: %d, Device ID: %s\n", lLoginID, pDeviceInfo->byDeviceID);

} //Input parameters
NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO )nBr
pServerInfo->dwTimeOutCount = 6; //Heartbeat timeout
times pServerInfo->dwKeepAliveSec = 15; //Heartbeat interval

} else if (ENUM_DEV_OFF ==

dwDataType) { rntf(n lUserID: %d\n", lUserID);
NET_ECMS_ForceLogout(lUserID); } else

{ }

return TRUE;
}

/// ...

BOOL InputStreamData(BYTE byDataType, char* Br int iDataLen) {

(IN      == NULL)
{
V = fopen("Test.mp4","wb"); rntf(v data to
Test.mp4!");
}

(IN      NULL)
{
wr(Brn1V) //Download video file
}

//Call the playback library to decode and display the code stream to

achieve playback/* if(1 ==
byDataType) {
if (!PlayM4_GetPort(&m_lPort))
{
return FALSE;
}

} if (!PlayM4_SetStreamOpenMode(m_lPort, STREAME_REALTIME))
{
return FALSE;
}
//Enter r 40-byte header if!
PlayM4_OpenStream(m_lPort, (unsigned char )Br (DWORD)iDataLen, 2*1024*1024)) {

return FALSE;
} if(IPayM4_Play(m_lPort, hWnd))

```

```

    {
        return FALSE;
    }

}

else { for (int i=0; i<1000; i++) {

    BOOL bRet = PlayM4_InputData(m_lPort,(unsigned char )Br (DWORD)iDataLen); if (!
    bRet) {

        if ( i >=999) {

            rntf(yM4n           failed, error code: %d!", PlayM4_GetLastError(m_lPort));
        }
        Sleep(2);

    } else

    { break; } }

}

}
/* return TRUE;
}

}

/// ...

BOOL CALLBACK fnPLAYBACK_DATA_CB(LONG IPlayBackLinkHandle, NET_EHOME_PLAYBACK_DATA_CB_INFO
*pDataCBInfo, void *pUserData) {

if (NULL == pDataCBInfo)
{
    return FALSE;

} IPlayHandle = IPlayBackLinkHandle;
InputStreamData(pDataCBInfo->dwType, (char*)pDataCBInfo->pData, pDataCBInfo->dwDataLen); return
TRUE;
}

}

/// ...

BOOL CALLBACK fnPLAYBACK_NEWLINK_CB(LONG IPlayBackLinkHandle, NET_EHOME_PLAYBACK_NEWLINK_CB_INFO
*pNewLinkCBMsg, void *pUserData) {

ILinkHandle = IPlayBackLinkHandle;
rntf(Cbc of playback listening, Device ID: %s\n", pNewLinkCBMsg->szDeviceID);

//Callback parameters of
playback data NET_EHOME_PLAYBACK_DATA_CB_PARAM
struDataCB = {0}; struDataCB.fnPlayBackDataCB = fnPLAYBACK_DATA_CB;

```

```

struDataCB.byStreamFormat = 0; //Encapsulation format: 0-PS format

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

} rntf(NRMyBcCBn)

return TRUE;
}

/// ...

NRMxctinCbc(0 0, rmxctinCbc NULL);

void CALLBACK rmxctinCbc(WR dwType, LONG iUserID, LONG iHandle, void* pUser) {

    if(EHOME_PREVIEW_EXCEPTION == dwType)
    {
        //Preview exception
        rntf(rvw xctin handle=%d, Error:%d", iHandle, dwError);

    } else if (EHOME_PLAYBACK_EXCEPTION == dwType)
    {
        // Playback exception
        rntf(ybc xctin handle=%d, Error:%d", iHandle, dwError);

    } else if (EHOME_AUDIOTALK_EXCEPTION == dwType)
    {
        //Voice talk (forwarding)
        exceptionrntf(rm VoiceTalk xctin handle=%d, Error:%d", iHandle, dwError);
    }
}

void main()
{
    //SMS gets the code stream after starting the monitoring service

    // Initialize SMS library
    NET_ESTREAM_Init();

    //Listening parameters
    for playback 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
    listening port number struListen.fnNewLinkCB = fnPLAYBACK_NEWLINK_CB; //Callback
    function for playback request
    struListen.pUserData = NULL; struListen.byLinkMode = 0; //0-TCP, 1-UDP (reserved)

    //Start monitoring service
}

```

```
LONG IHandle = NET_ESTREAM_StartListenPlayBack(&struListen);
if(IHandle < -1)
{
    rntf(NRMrnyBc failed, error code: %d\n", NET_ESTREAM_GetLastError());
    NET_ESTREAM_End();
    return

} rntf(NRMrnyBcn) //////////////////////

/// ...

// Initialize CMS library
NET_ECMS_Init();

//Registered listening
parameters 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;

//Start monitoring service and receive device registration information
LONG IListen = NET_ECMS_StartListen(&struCMSListenPara);
if(IList < -1) {

    rntf(NCMrn failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMrnn)

while(1)
{
    Sleep(1000); //The following operations must be performed after registration is
    completed if(LoginID >=
    0) {
        break;
    }
}

//Search for video
files NET_EHOME_REC_FILE_COND struFindCond =
{0}; struFindCond.dwChannel = 1; //Channel number, starting
from 1 struFindCond.dwRecType = 0x //All types
struFindCond.dwStartIndex = 0; //Search starting position
struFindCond.dwMaxFileCountPer = 5; //Maximum number of files for a single search

//Find the start
time 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;

//ÿÿÿÿÿ
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 ISearchType = 0; //Search for files
LONG IFileHandle = NET_ECMS_StartFindFile_V11(ILoginID, ISearchType, &struFindCond, sizeof(struFindCond)); //Start search if
(IFileHandle < 0) {

    rntf(NCMrFnFV11 failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

} rntf(NCMrFnFV11n)

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

NET_EHOME_REC_FILE struFileInfo = {0};

//You can search for files in the
thread
while(1) {
    IRet = NET_ECMS_FindNextFile_V11(IFileHandle, &struFileInfo, sizeof(struFileInfo)); //Get search results one by one
    if (IRet == ENUM_GET_NEXT_STATUS_SUCCESS)
    {
        if (struFileInfo.dwFileSize / 1024 == 0) {

            rntf(cmrFnwF)

        } else if (struFileInfo.dwFileSize / 1024 > 0 && struFileInfo.dwFileSize /(1024*1024) == 0) {

            rntf(cmrFnwF1024)

        } else
        {
            rntf(cmMrFnwF /1024/1024);
        }

        rntf(Fnm %02d:      Filesize[%s], StarTime[%04d-%02d-%02d %02d:%02d:%02d], StopTime[%04d-%02d-%02d
%02d:%02d] \n", \
struFileInfo.szFileName, csTmp, struFileInfo.struStartTime.wYear, struFileInfo.struStartTime.byMonth, \

```

```

struFileInfo.struStartTime.byDay, struFileInfo.struStartTime.byHour, struFileInfo.struStartTime.byMinute,\n
struFileInfo.struStartTime.bySecond, struFileInfo.struStopTime.byDay, struFileInfo.struStopTime.byHour, \
struFileInfo.struStopTime.byMinute, struFileInfo.struStopTime.bySecond); //Searched nrmtin

memcpy(szFileName, struFileInfo.sFileName, MAX_FILE_NAME_LEN);

} else
{
    if (lRet == ENUM_GET_NETX_STATUS_NEED_WAIT) {

        Sleep(5);
        cntin
    }

    if ((lRet == ENUM_GET_NETX_STATUS_NO_FILE) || (lRet == ENUM_GET_NEXT_STATUS_FINISH)) {

        rntf(N more      n)
        break;
    }

    } else if(lRet == ENUM_GET_NEXT_STATUS_NOT_SUPPORT) {

        rntf(vc does not support\n); break;

    } else
    {
        rntf(F      to na      for the server is busy or network failure\n);
        break;
    }
}

}

//Input parameters of
playback request NET_EHOME_PLAYBACK_INFO_IN
struPlayBackIn = {0}; struPlayBackIn.dwSize =
sizeof(struPlayBackIn); struPlayBackIn.dwChannel
= 1; //Channel number struPlayBackIn.byPlayBackMode = 0; //Playback mode: 0-playback by file name, 1-
playback by time (reserved) struPlayBackIn.unionPlayBackMode.struPlayBackByName.dwSeekType = 0; //0-calculated by byte
length, 1-calculated by seconds ryBcnnyBcMryBcbNmwF = 0; //When dwSeekType
is 0, the offset is calculated in bytes; when dwSeekType is 1, the offset is calculated in seconds
struPlayBackIn.unionPlayBackMode.struPlayBackByName.dwFileSpan = 0; //The size of the downloaded file. If the size is 0, it means
The file has been downloaded
memcpy(struPlayBackIn.unionPlayBackMode.struPlayBackByName.szFileName, szFileName,
MAX_FILE_NAME_LEN); //The name of the file to be played back

memcpy(struPlayBackIn.struStreamSever.szIP, "10.16.2.123", sizeof("10.16.2.123")); //SMS IP address
struPlayBackIn.struStreamSever.wPort = 8003; //SMS port number, must be consistent with the listening port number

//Output parameters of playback request
NET_EHOME_PLAYBACK_INFO_OUT struPlayBackOut = {0};

```

```

//Playback
request if(!NET_ECMS_StartPlayBack(lLoginID, &struPlayBackIn, &struPlayBackOut)) {

    rntf(NCMryBc failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMryBcn)

//Input parameters of the code
stream transmission request NET_EHOME_PUSHPLAYBACK_IN
struPushPlayBackIn = {0}; struPushPlayBackIn.dwSize =
sizeof(struPushPlayBackIn); struPushPlayBackIn.lSessionID = struPlayBackOut.lSessionID; //Session ID of the playback request

// Output parameters of code stream transmission request
NET_EHOME_PUSHPLAYBACK_OUT struPushPlayBackOut = {0};

//Send a request to the device and start
streaming if(NET_ECMS_StartPushPlayBack(lLoginID, &struPushPlayBackIn, &struPushPlayBackOut)) {

    rntf(NCMryBc failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMryBcn)

Sleep(50000);

/////////////////////////////// //Sign out

//CMS stops listening
service if(!NET_ECMS_StopListen(lListen))
{
    rntf(NCMn failed, error code: %d\n", NET_ECMS_GetLastError());
}

//Release the resources occupied by CMS

NET_ECMS_Fini(); //////////////////////

//SMS stops forwarding
stream if(lPlayHandle
>= 0) {
    if (!NET_ESTREAM_StopPlayBack(lPlayHandle)) {

        rntf(NRMyBc failed, error code: %d\n", NET_ECMS_GetLastError());
    }
}

//SMS stops monitoring service

```

```
if(ILinkHandle >= 0) {  
  
    if (!NET_ESTREAM_StopListenPlayBack(ILinkHandle)) {  
  
        rntf(NRMnyBc failed, error code: %d\n", NET_ECMS_GetLastError());  
    }  
  
    } //Release the resources occupied by SMS  
    NET_ESTREAM_End();  
  
    //Release file resources  
    (V != NULL) {  
  
        c(V)  
        IN      = NULL;  
    }  
  
    rntf(xn)  
}
```

Subsequent

processing • Call NET_ECMS_StopListen to stop the CMS listening service. • Call NET_ECMS_Fini and NET_ESTREAM_Fini to release the resources occupied by CMS and SMS.

Chapter 6 Voice Intercom

6.1 Enable voice intercom

The voice intercom function enables the sending and receiving of audio data between the monitoring center and the specified device. It is usually used in the following scenarios: when there are multiple entrances and exits in a park or campus, and cameras are installed at the booths at each entrance and exit, if an abnormality occurs at a certain entrance and exit, the monitoring center personnel can use the voice intercom function to talk to the on-duty personnel in the corresponding booth to understand the specific situation and issue processing or command commands. Here, the voice intercom is implemented by the Center Management Service (CMS) and the process is only applicable to devices that support ISUP2.0.

Prerequisites •

Make sure that *NET_ECMS_Init* has been called to initialize CMS. • Make sure that *NET_ECMS_StartListen* has been called to start the CMS listening service in order to receive device registration information.

Steps

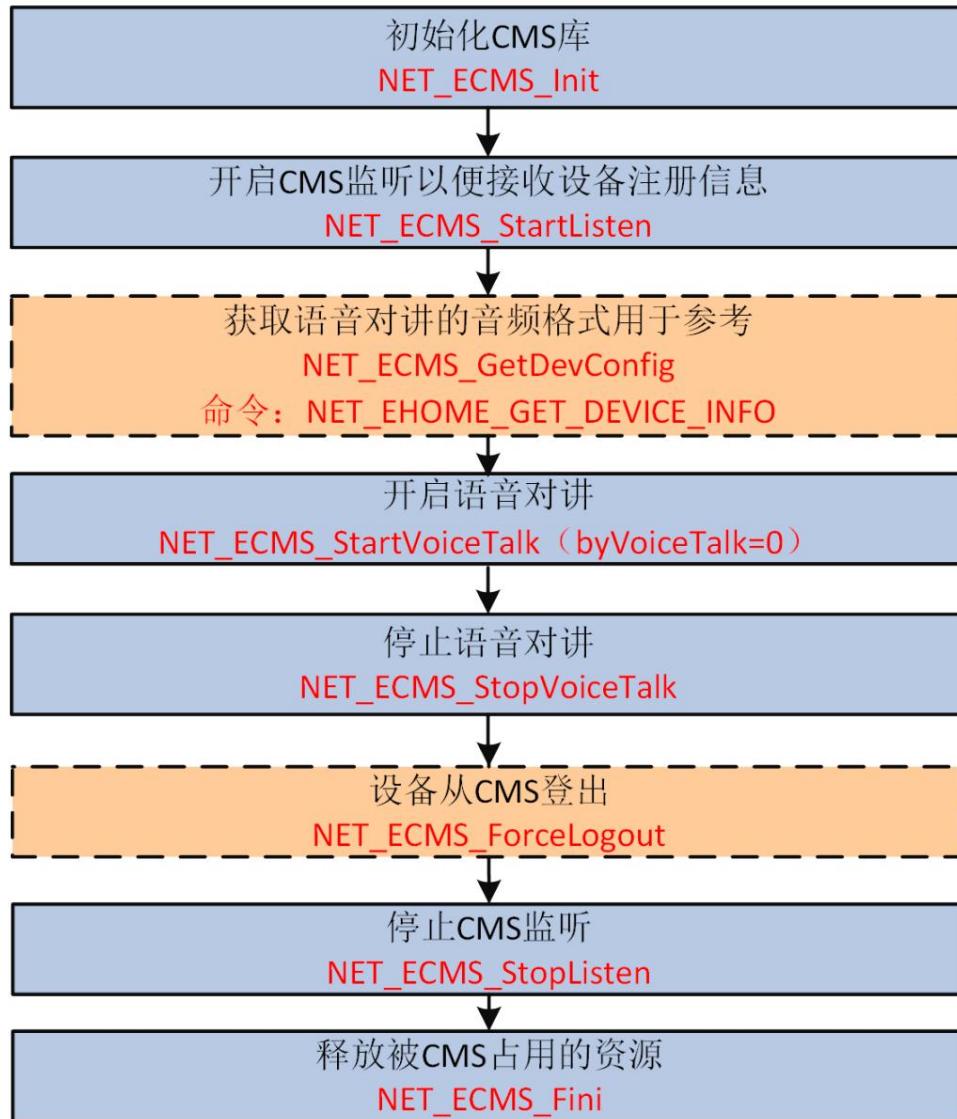


Figure 6-1 Flowchart of the interface call to enable voice intercom



In order to determine the ISUP version information supported by the device, you can set **the dwDataType** in the callback function **DEVICE_REGISTER_CB** to "ENUM_DEV_ON" when calling **NET_ECMS_StartListen** to start CMS listening and register the callback function. The protocol version information supported by the device will be called back through the member **byDevProtocolVersion** in the structure **NET_EHOME_DEV_REG_INFO**.

1. Optional operation: Call **NCMGCn** and "**NET_EHOME_GET_DEVICE_INFO**" (command number: 1) to obtain the audio formats supported by voice intercom for reference.

The supported audio formats are returned by **pOutBuf** through the *NET_EHOME_DEVICE_INFO* structure .

2. Call *NET_ECMS_StartVoiceTalk* and set **byVoiceTalk** in *NET_EHOME_VOICETALK PARA* to

"0" to start voice intercom. 3. Call

NET_ECMS_StopVoiceTalk to stop voice forwarding and intercom. 4. Optional

operation: Call *NET_ECMS_ForceLogout* to force logout of the device.

Example code to enable voice intercom

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

LONG ILoginID = -1;
LONG m_lVoiceTalkHandle = -1;

/// ...

BOOL CALLBACK RegisterCallBack(LONG lUserID, DWORD dwDataType, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDeviceInfo = (NET_EHOME_DEV_REG_INFO *)pUser

        if (pDeviceInfo != NULL)

            { lUserID = lUserID;
              rntf(nn lUserID: %d, Device ID: %s\n", lUserID, pDeviceInfo->byDeviceID);

        } //Input parameters
        NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO *)pUser
        pServerInfo->dwTimeOutCount = 6; //heartbeat timeout
        times pServerInfo->dwKeepAliveSec = 15; //

        heartbeat interval} else if (ENUM_DEV_OFF

        == dwDataType) { rntf(n lUserID: %d\n", lUserID);
        NET_ECMS_ForceLogout(lUserID); } else

        {}

    }

    return TRUE;
}

////////////////////////////// //Voice data
callback function void
CALLBACK g_fVoiceDataCallBack(LONG iVoiceComHandle, char RcvBr DWORD dwBufSize, DWORD
```

ISUPSDK (General) Development Guide

```
dwEncodeType, BYTE byAudioFlag, void *pUser) { //Get the
voice data and process it yourself}

void main(){

///////////////////////////////
//CMS registration and preview request

//CMS registration module initialization
NET_ECMS_Init();

//Register listening
parameters 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;

//Start the monitoring service and receive device registration information
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara); if(lListen < -1) {

    rntf(NCMrn failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMrnn)

while(1 {

    Sleep(1000); //Perform subsequent operations after successful
    registration if(lLoginID
    >= 0) {
        break;
    }
}

//Voice talk parameters
NET_EHOME_VOICETALK_PARA struVoiceTalkPara = {0};
struVoiceTalkPara.bNeedCBNoEncData = 0; //Voice type that needs callback: 0- encoded voice, 1- encoded voice (not supported when voice forwarding)

struVoiceTalkPara.cbVoiceDataCallBack = g_fVoiceDataCallBack;
struVoiceTalkPara.dwEncodeType = 0; //Voice encoding type of the device: 0- G722, 1- G711U, 2- G711A, 3- G726, 4- AAC,
5- MP2L2ÿ6- PCM
struVoiceTalkPara.byVoiceTalk = 0; //Voice talk mode: 0- voice talk, 1- voice forwarding

DWORD dwVoiceChan = 1; //Voice intercom channel number

//Start voice intercom
```

```

m_lVoiceTalkHandle = NET_ECMS_StartVoiceTalk(lLoginID, dwVoiceChan, &struVoiceTalkPara); if
(m_lVoiceTalkHandle < 0) {

    rntf(NCMrVc failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

Sleep(30000);

//Stop voice talkif(
NET_ECMS_StopVoiceTalk(m_lVoiceTalkHandle)) {

    rntf(NCMVc failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

/////////////////////////////// //quit

//CMS stops listening
service if(!NET_ECMS_StopListen(lListen))
{
    rntf(NCMn failed, error code: %d\n", NET_ECMS_GetLastError());
}

//CMS deinitialization, release resources
NET_ECMS_Fini();

rntf(xn)
}

```

Subsequent

processing • Call NET_ECMS_StopListen to stop the CMS listening service. •

Call NET_ECMS_Fini to release the resources occupied by CMS.

6.2 Voice forwarding

If there are multiple entrances and exits in a park or campus, and cameras are installed in the corresponding booths of each entrance and exit, the monitoring center can send audio information to a designated camera for conversation through the voice forwarding function as needed. Here, the voice forwarding function is implemented through the central management service (CMS).

Prerequisites •

Make sure that NET_ECMS_Init has been called to initialize CMS. • Make

sure that NET_ECMS_StartListen has been called to start the CMS listening service in order to receive device registration information.

Steps

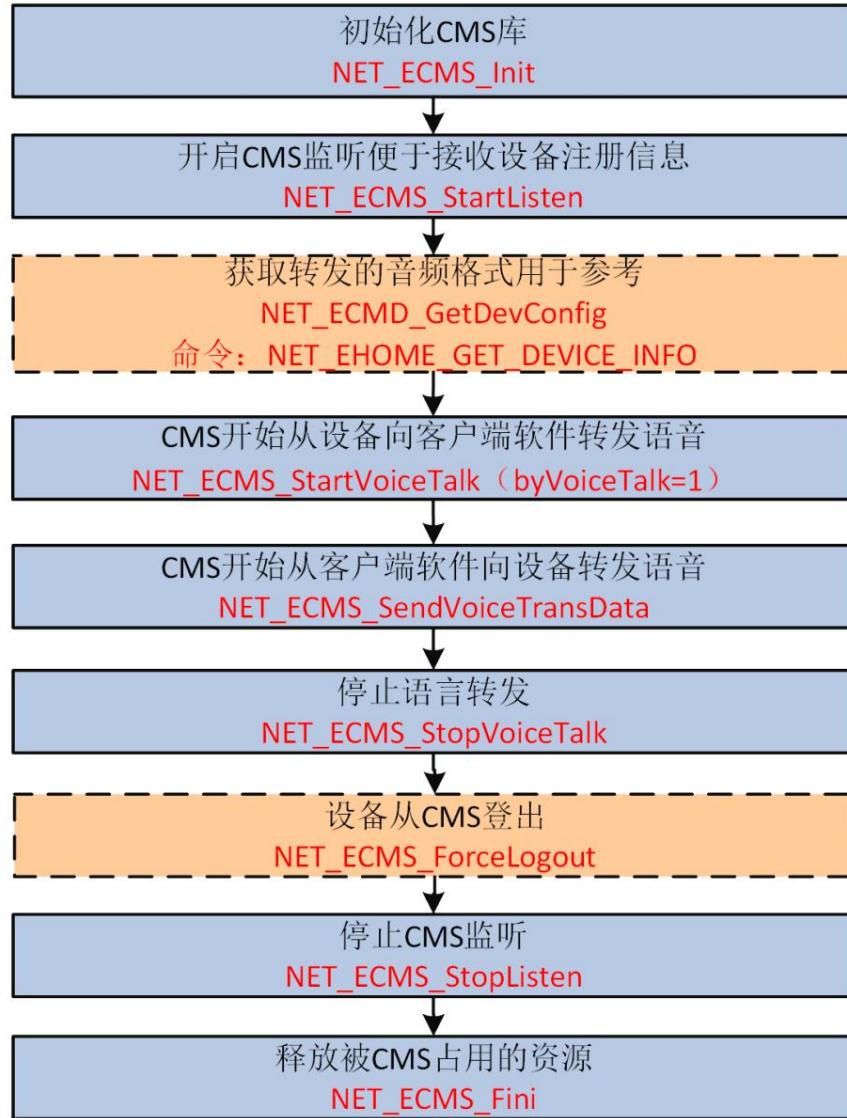


Figure 6-2 Interface call flow chart for starting voice

forwarding 1. Optional operation: Call `NCMGCh` and "NET_EHOME_GET_DEVICE_INFO" (command number: 1) to obtain the audio formats supported by voice intercom for reference.

The supported audio formats are returned by `pOutBuf` through the structure `NET_EHOME_DEVICE_INFO`.

2. Call `NET_ECMS_StartVoiceTalk` and set `byVoiceTalk` in `NET_EHOME_VOICETALK_PARA` to "1" to forward the voice from the device to the client software. 3.

Call `NET_ECMS_SendVoiceTransData` to forward the voice from the client software to the device. 4.

Call `NET_ECMS_StopVoiceTalk` to stop voice forwarding and intercom.

5. Optional operation: Call `NET_ECMS_ForceLogout` to force the device to log out.

Subsequent

processing • Call `NET_ECMS_StopListen` to stop the CMS listening service. • Call `NET_ECMS_Fini` to release the resources occupied by CMS.

6.3 Enable voice intercom via streaming media service

The voice intercom function enables the sending and receiving of audio data between the monitoring center and the designated device. It is usually used in the following scenarios: when there are multiple entrances and exits in a park or campus, and cameras are installed at the booths at each entrance and exit, if an abnormality occurs at a certain entrance and exit, the monitoring center personnel can use the voice intercom function to talk to the on-duty personnel in the corresponding booth to understand the specific situation and issue processing or command commands. Here, the voice intercom is implemented by the streaming media service (SMS) and the center management service (CMS), and the process is applicable to devices that support ISUP4.0 and ISUP5.0.

Prerequisites •

Make sure that `NET_ECMS_Init` and `NET_ESTREAM_Init` have been called to initialize CMS and SMS. • Make sure that `NET_ECMS_StartListen` has been called to start the CMS listening service in order to receive device registration information.

Steps

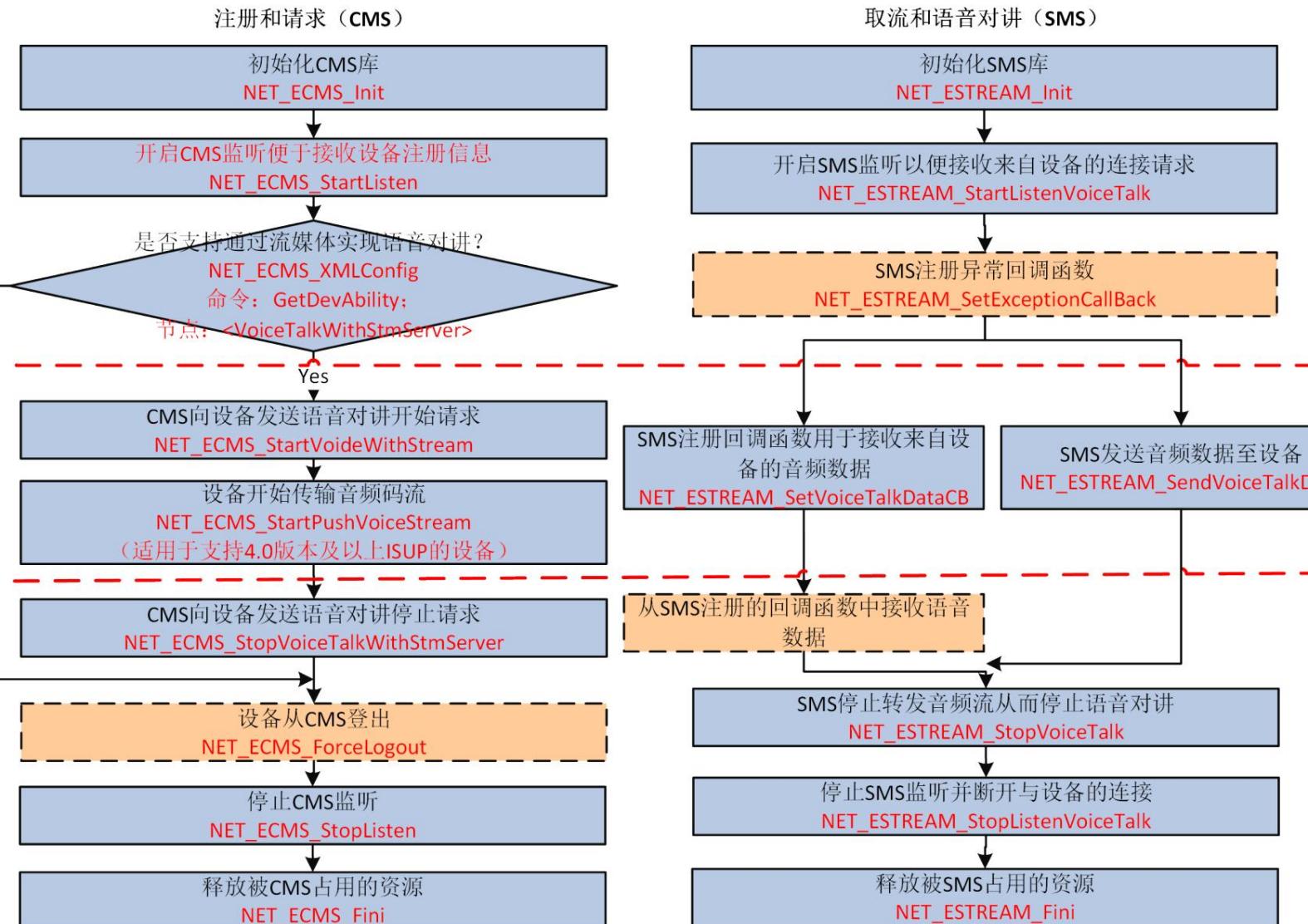


Figure 6-3 Flowchart of the interface call to enable voice intercom through streaming media service



In order to determine the ISUP version information supported by the device, you can set the `dwDataType` in the callback function `DEVICE_REGISTER_CB` to "ENUM_DEV_ON" when calling `NET_ECMS_StartListen` to start CMS listening and register the callback function. The protocol version information supported by the device will be called back through the member `byDevProtocolVersion` in the structure `NET_EHOME_DEV_REG_INFO`.

1. Call `NCMMCn` and command `GetDevAbility` to obtain `device capabilities` and determine whether the device supports streaming. The media service enables voice intercom.

ISUPSDK (General) Development Guide

The device capability set is returned by **pOutBuf** .

If supported, the node <VoiceTalkWithStmServer> will be returned and you can continue with the following steps.

Otherwise, end the task.

2. Call NET_ESTREAM_StartListenPreview to start the SMS listening service and receive the connection request from the device. 3. Optional operation: Call NRMxcnCBc to register the exception callback function for SMS. 4. Call NET_ECMS_StartVoiceWithStmServer to send the voice intercom start request from CMS to the device.

Send the SMS address and port number to the device, and the device automatically assigns a session ID to the CMS. 5. Call

NET_ECMS_StartPushVoiceStream to send the voice transmission request from the CMS to the device.

The device automatically connects to SMS and starts sending audio data to SMS

6. Perform one of the following operations to receive audio data from the device or send audio data to the device. - a. Call NET_ESTREAM_SetVoiceTalkDataCB to register a callback function for SMS to receive audio data from the device.

Frequency data.

b. Get the audio data from the voice callback function registered by SMS and forward it to the client software.



The decoding and display of the voice intercom audio code stream is implemented by the client

software. -Call NET_ESTREAM_SendVoiceTalkData to send the audio data to the device.

7. Call NET_ESTREAM_StopVoiceTalk to stop SMS audio stream forwarding. 8. Call

NET_ECMS_StopVoiceTalkWithStmServer to send the voice talk stop request from CMS to the device.

The device stops transmitting the audio code

stream. 9. Call NET_ESTREAM_StopListenVoiceTalk to stop the SMS listening service and disconnect it from the device. 10. Optional operation: Call

NET_ECMS_ForceLogout to force the device to log out of CMS.

Example

Sample code for starting voice intercom via streaming media service



In this sample code, CMS and SMS are installed in the same application on the same computer.

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

LONG ILoginID = -1;
LONG IVoiceLinkHandle = -1;
LONG IVoiceHandle = -1;
FILE          = NULL;

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

```

//Register callback function
BOOL CALLBACK RegisterCallBack(LONG lUserID, DWORD dwDataType, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDeviceInfo = (NET_EHOME_DEV_REG_INFO )Br
        if (pDeviceInfo != NULL)
        {
            ILoginID = lUserID;
            rntf(nn lUserID: %d, Device ID: %s\n", ILoginID, pDeviceInfo->byDeviceID);

            } //Input parameters
            NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO )nBr
            pServerInfo->dwTimeOutCount = 6; //Heartbeat timeout
            times pServerInfo->dwKeepAliveSec = 15; //Heartbeat interval

        } else if (ENUM_DEV_OFF ==
        dwDataType) { rntf(n lUserID: %d\n", lUserID);
        NET_ECMS_ForceLogout(lUserID); } else
        {}

        return TRUE;
    }

    /// ...

    BOOL InputVoiceData(char* Br int iDataLen) {
        == NULL)
        {
            = fopen("Test.mp4","wb");
            rntf(v data to Test.mp4!\n");
        }

        NULL)
        {
            wr(Brn1) //Call back the real-time stream to directly write the file and save the video
        }

        //.....Other data processing, voice data forwarding, decoding and playback need to be implemented by yourself

        return TRUE;
    }

    //////////////////////////////////////////////////////////////////

```

```

//Voice data callback
function void CALLBACK fnVOICETALK_DATA_CB(LONG IHandle, NET_EHOME_VOICETALK_DATA_CB_INFO *pDataCBInfo, void *pUserData)
{
    if (NULL == pDataCBInfo) {
        return ;
    }
    IVoiceHandle = IHandle;
    InputVoiceData((char*)pDataCBInfo->pData, pDataCBInfo->dwDataLen);
}

////////////////////////////// //SMS voice intercom request response callback function
BOOL CALLBACK fnVOICETALK_NEWLINK_CB(LONG ILinkHandle, NET_EHOME_VOICETALK_NEWLINK_CB_INFO *pNewLinkCBInfo,
void *pUserData) {
    IVoiceLinkHandle = ILinkHandle; rntf(Cbc
        of two-way audio listening, Device ID: %s, Audio Channel: %d, Audio Type: %d\n", pNewLinkCBInfo->szDeviceID, pNewLinkCBInfo->dwAudioChan, pNewLinkCBInfo->dwEncodeType);

    //Voice data callback
    parameters NET_EHOME_VOICETALK_DATA_CB_PARAM struDataCB =
    {0}; struDataCB.fnVoiceTalkDataCB = fnVOICETALK_DATA_CB;

    if (!NET_ESTREAM_SetVoiceTalkDataCB(IVoiceLinkHandle, &struDataCB)) {
        rntf(NRMVcCB failed, error code: %d\n", NET_ESTREAM_GetLastError());
        return FALSE;
    }
    rntf(NRMVcCBn)

    return TRUE;
}

////////////////////////////// //Read the voice data from
the file and send it to the device
DWORD WINAPI TestVoiceTransThread(LPVOID lpArg) {
    LONG IHandle = (LONG)lpArg;

    FILE *pFile = fopen("ehome_send.audio", "rb"); //Open the local audio file. The audio format must be consistent with the device audio format if (pFile
== NULL) {
        return 0;
    }

    char szBuff[80] = {0}; //Take G722 audio type as an example, 80 bytes of data are sent each
    time int nRemainLen = 0;

    int nCurPos = fseek(pFile, SEEK_END);

```

```
nRemainLen = ft(F) fseek(pFile,
nCurPos, SEEK_SET);

int nSendTotal = 0;
while(1) {

    if (nRemainLen >= 80) {

        nSendTotal += 80;
        fread(szBuf, 80, 1, pFile);

        Voice talk data parameters
        NET_EHOME_VOICETALK_DATA struVoicTalkData = {0};
        struVoicTalkData.pSendBuf = (BYTE*)szBuf;
        struVoicTalkData.dwDataLen = 80;

        NET_ESTREAM_SendVoiceTalkData(lHandle, &struVoicTalkData); nRemainLen -= 80;

    } else

    {

        nSendTotal += nRemainLen;
        fread(szBuf, nRemainLen, 1, pFile);

        Voice talk data parameters
        NET_EHOME_VOICETALK_DATA struVoicTalkData = {0};
        struVoicTalkData.pSendBuf = (BYTE*)szBuf;
        struVoicTalkData.dwDataLen = nRemainLen;

        NET_ESTREAM_SendVoiceTalkData(lHandle, &struVoicTalkData); break;

    }

    Sleep(15);
}

fclose(pFile);
return 0;
}

void main()
{
    //SMS start service

    //SMS initialization
    NET_ESTREAM_Init();

    //Voice talk monitoring
    parameters NET_EHOME_LISTEN_VOICETALK_CFG struListen = {0};
    memcpy(struListen.struIPAdress.szIP, "10.16.2.123", sizeof("10.16.2.123")); struListen.struIPAdress.wPort
    = 8003; //Streaming media service listening port struListen.fnNewLinkCB =
    fnVOICETALK_NEWSLINK_CB; //Voice talk connection request callback function struListen.pUser = NULL;
```

```

//Start voice intercom monitoring
LONG lHandle = NET_ESTREAM_StartListenVoiceTalk(&struListen); if(lHandle <
-1) {

    rntf(NRMrnVc failed, error code: %d\n", NET_ESTREAM_GetLastError());
    NET_ESTREAM_End();
    return

} rntf(NRMrnVcn) /////////////////
///////////////////////////////
//CMS registration and voice intercom request

//CMS registration module initialization
NET_ECMS_Init();

//Register listening
parameters 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;

//Start the monitoring service and receive device registration information
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara); if(lListen < -1)
{

    rntf(NCMrn failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMrn)

while(1) {

    Sleep(1000); //Perform follow-up operations after successful
    registration if(lLoginID
    >= 0) {
        break;
    }
}

//Voice talk request input
parameter NET_EHOME_VOICE_TALK_IN struVoiceTalkIn =
{0}; struVoiceTalkIn.dwVoiceChan = 1; //Voice talk channel number
memcpy(struVoiceTalkIn.struStreamSever.szIP, "10.16.2.123", sizeof("10.16.2.123")); //Streaming media service IP address
struVoiceTalkIn.struStreamSever.wPort = 8003; //Streaming media service port, which needs to be consistent with the service listening port

//Voice intercom request output parameters
NET_EHOME_VOICE_TALK_OUT struVoiceTalkOut = {0};

```

```

//CMS voice intercom
request if(!NET_ECMS_StartVoiceWithStmServer(lLoginID, &struVoiceTalkIn, &struVoiceTalkOut)) {

    rntf(NCMrVcWmrvr failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

} rntf(NCMrVcWmrvrn)

//CMS requests to start streaming
NET_EHOME_PUSHVOICE_IN struPushVoiceln = {0};
struPushVoiceln.dwSize = sizeof(struPushVoiceln);
struPushVoiceln.lSessionID = struVoiceTalkOut.lSessionID;
NET_EHOME_PUSHVOICE_OUT struPushVoiceOut = {0};
struPushVoiceOut.dwSize = sizeof(struPushVoiceOut); if (!
NET_ECMS_StartPushVoiceStream(lLoginID, &struPushVoiceln, &struPushVoiceOut)) {

    rntf(NCMrVcrm failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;
}

} rntf(NCMrVcrmn)

//Create a thread to read voice data from the file and send it to the device
DWORD dwThreadId;
CreateThread(NULL, 0, LPTHREAD_START_ROUTINE(TestVoiceTransThread), (void*)lVoiceLinkHandle, 0,
&dwThreadId);

Sleep(50000);

/////////////////////////////// //quit

//Release CMS voice intercom request
resources if(!NET_ECMS_StopVoiceTalkWithStmServer(lLoginID, struVoiceTalkOut.lSessionID)) {

    rntf(NCMVcWmrvr failed, error code: %d\n", NET_ECMS_GetLastError());
}

//CMS stops listening
service if(!NET_ECMS_StopListen(lListen))
{
    rntf(NCMn failed, error code: %d\n", NET_ECMS_GetLastError());
}

//CMS deinitialization, release resources

NET_ECMS_Fini(); //////////////////////

//SMS stops voice intercom

```

```
if(!VoiceHandle >= 0)
{
    if (!NET_ESTREAM_StopVoiceTalk(!VoiceHandle)) {

        rntf(NRMVc failed, error code: %d\n", NET_ECMS_GetLastError());
    }
}

//SMS stops voice intercom monitoring
service if(!VoiceLinkHandle >=
0) {
    if (!NET_ESTREAM_StopListenVoiceTalk(!VoiceLinkHandle)) {

        rntf(NRMnVc failed, error code: %d\n", NET_ECMS_GetLastError());
    }
}

//SMS deinitialization, release resources
NET_ESTREAM_End();

//Release file resources!
    = NULL)
{{

    c()
    = NULL;
}

    rntf(xn)
}
```

Subsequent

processing • Call *NET_ECMS_StopListen* to stop the CMS listening service. • Call *NET_ECMS_Fini* and *NET_ESTREAM_Fini* to release the resources occupied by CMS and SMS.

Chapter 7 PTZ Control

PTZ control enables the camera to pan, tilt and zoom. You can set preset points for a specific surveillance area to quickly switch the camera screen as needed or reposition the selected area to the center of the screen through 3D positioning.

Control PTZ	Call <i>NET_ECMS_RemoteControl</i> and set dwCommand to "NET_EHOME_PTZ_CTRL" (Command No.: 1000), and nb in <i>NET_EHOME_REMOTE_CTRL_PARAM</i> is set to <i>NET_EHOME_PTZ_PARAM</i> \downarrow
To configure, call, and clear presets, call <i>NET_ECMS_RemoteControl</i> and set dwCommand to "NET_EHOME_PRESET_CTRL" (command number: 1001), and nb in <i>NET_EHOME_REMOTE_CTRL_PARAM</i> is set to <i>NET_EHOME_PRESET_PARAM</i> \downarrow	
Perform 3D positioning	Call <i>NCMMCn</i> to transmit command message <i>PZIN</i> \downarrow

Chapter 8 Alarm or Event Configuration

This section shows the types of alarms or events that can be configured through ISUPSDK. Before triggering and receiving alarms or events, you can

Set alarm or event parameters such as rules, deployment plans, and alarm linkage behaviors.



illustrate

If alarm or event parameters are configured, the alarm listening service can be enabled to receive device notifications when an alarm is triggered or an event occurs.

Automatically upload alarm or event information (see for details). ~~For alarm monitoring, please enable motion detection to receive alarms.~~

See 16-1 and

Event type and details

④

Motion Detection

Table 8-1 Motion detection

Get motion detection parameters	Call <code>NCMMCn</code> to transmit command message <code>GMn</code> .	④
Configuring Motion Detection Parameters	Call <code>NCMMCn</code> to transmit command message <code>Mn</code>	④
To obtain the motion detection area parameters, call <code>NCMMCn</code> to transmit the command message <code>GMn</code> .		④
Configure motion detection area parameters and call <code>NCMMCn</code> to transmit command message <code>Mn</code>		④

Video occlusion detection

Table 8-2 Video occlusion detection

To obtain the video occlusion detection parameters, call <code>NCMMCn</code> to transmit the command message <code>GetHideAlarmPara</code> .	④
To configure the video occlusion detection parameters, call <code>NCMMCn</code> to transmit the command message <code>SetHideAlarmPara</code> .	④
Get the video occlusion detection area parameter	Call <code>NCMMCn</code> to transmit the command message <code>GetHideAlarmArea</code> .
Configure video occlusion detection area parameter	Call <code>NCMMCn</code> to transmit the command message <code>SetHideAlarmArea</code> .

Video Loss Detection

Table 8-3 Video loss detection

To obtain the video loss detection parameters, call the <u>NCMMCn</u> transmission command message <u>GetVILostPara</u> .	_____
To set the video loss detection parameters, call <u>NCMMCn</u> to transmit the command message <u>SetVILostPara</u> .	_____

Privacy Masking

Table 8-4 Privacy Masking

Get privacy mask parameters	Call <u>NCMMCn</u> to transmit the command message <u>GetHidePara</u> .	_____
Configuring privacy masking parameters	Call <u>NCMMCn</u> to transmit the command message <u>SetHidePara</u> .	_____
To obtain the privacy mask area parameters, call <u>NCMMCn</u> to transmit the command message <u>GetHideArea</u> .	_____	_____
To configure the privacy mask area parameters, call <u>NCMMCn</u> to transmit the command message <u>SetHideArea</u> .	_____	_____

Abnormal alarm

Table 8-5 Abnormal alarm

Get abnormal alarm parameters	Call <u>NCMMCn</u> to transmit command message <u>Gxcn</u> .	_____
Configure abnormal alarm parameters	Call <u>NCMMCn</u> to transmit command message <u>xcn</u> .	_____

Passenger flow statistics

Table 8-6 Passenger flow statistics

Get passenger flow statistics parameters	Call <u>NCMMCn</u> to transmit the command message <u>GetPassengerPara</u> .	_____
Set passenger flow statistics parameters	Call <u>NCMMCn</u> to transmit the command message <u>SetPassengerPara</u> .	_____
Get passenger flow statistics information OSD parameters	Call <u>NCMMCn</u> transmission command message <u>GetPassengerOSDPara</u> .	_____
Configure passenger flow statistics OSD parameters	and call <u>NCMMCn</u> transmission command message <u>SetPassengerOSDPara</u> .	_____

ISUPSDK (General) Development Guide

Get the passenger flow statistics detection line parameters and call <u>NCMMCn</u> transmission command message	<u>GetPassengerLinePara</u> \circ
Configure the passenger flow statistics detection line parameters and call <u>NCMMCn</u> transmission command message	<u>SetPassengerLinePara</u> \circ
Get the calibration parameters of the passenger counting camera	Call <u>NCMMCn</u> to transmit command message <u>GCbn</u> \circ
Configuring the calibration parameters of the passenger counting camera	Call <u>NCMMCn</u> to transmit command message <u>Cbn</u> \circ

Alarm input

Table 8-7 Alarm input

Get alarm input parameters	Call <u>NCMGCn</u> with dwCommand set to "NET_EHOME_GET_ALARMIN_CFG" (command number: 11) and pCondBuf set to <u>NET_EHOME_ALARMIN_COND</u> . The alarm input parameters are returned by pOutBuf through the structure <u>NET_EHOME_ALARMIN_CFG</u> .
Get alarm output parameters	Call <u>NCMCn</u> and set dwCommand to "NET_EHOME_SET_ALARMIN_CFG" (Command No. 12), sets pCondBuf to <u>NET_EHOME_ALARMIN_COND</u> and pInBuf to <u>NET_EHOME_ALARMIN_CFG</u> \circ

Alarm Arming Plan

Table 8-8 Alarm deployment plan

Get the alarm deployment plan parameters and call <u>NCMMCn</u> to transmit command messages	<u>GetAlarmDeploymentTime</u> \circ
Configure alarm deployment plan parameters and call <u>NCMMCn</u> to transmit command messages	<u>SetAlarmDeploymentTime</u> \circ

Alarm linkage

Table 8-9 Alarm linkage

Get alarm linkage parameters	Call <u>NCMMCn</u> to transmit command message <u>GetAlarmLinkageType</u> ;
Configure alarm linkage parameters	Call <u>NCMMCn</u> to transmit command message <u>SetAlarmLinkageType</u> ;
Get snapshot linkage parameters	Call <u>NCMMCn</u> to transmit command message <u>GetAlarmTriggerCapture</u> ;
Configure snapshot linkage parameters	Call <u>NCMMCn</u> to transmit command message <u>SetAlarmTriggerCapture</u> ;
Get recording linkage parameters	Call <u>NCMMCn</u> to transmit command message <u>GetAlarmTriggerRecord</u> ;
Configure recording linkage parameters	Call <u>NCMMCn</u> to transmit command message <u>SetAlarmTriggerRecord</u> ;
Manually control the alarm output linkage and	call <u>NET_ECMS_RemoteControl</u> , set dwCommand to "NET_EHOME_MANUAL_IOOUT" (command number: 19), and nb in <u>NET_EHOME_REMOTE_CTRL_PARAM</u> is set to <u>NET_EHOME_MANUAL_IOOUT_CTRL</u> ;
Get alarm output parameters	Call <u>NCMGCn</u> and set dwCommand to "NET_ EHOME_GET_ALARMOUT_CFG" (command number: 15), and pCondBuf Set to a 4-digit alarm input number. The alarm output parameters are passed through the structure pOutBuf <u>NET_EHOME_ALARMOUT_CFG</u> is returned.
Configure alarm output parameters	Call <u>NCMCn</u> and set dwCommand to "NET_ EHOME_SET_ALARMOUT_CFG" (command number: 16), set pCondBuf Set to 4-bit alarm input number and set pInBuf to <u>NET_EHOME_ALARMOUT_CFG</u> ;

Chapter 9 : Enable monitoring service to receive alarms

Enabling the monitoring service and receiving device alarm information through ISUPSDK depends on the Central Management Service (CMS) and the Alarm Management Service (AMS). After the monitoring service is enabled, when the alarm is triggered, the device will automatically upload the alarm, so the alarm information can be received by configuring the monitoring service (ie AMS).

Prerequisites •

Make sure that `NET_ECMS_Init` and `NET_EALARM_Init` have been called to initialize CMS and AMS. • Make sure that `NET_ECMS_StartListen` has been called to start CMS listening to receive device registration information.

Steps

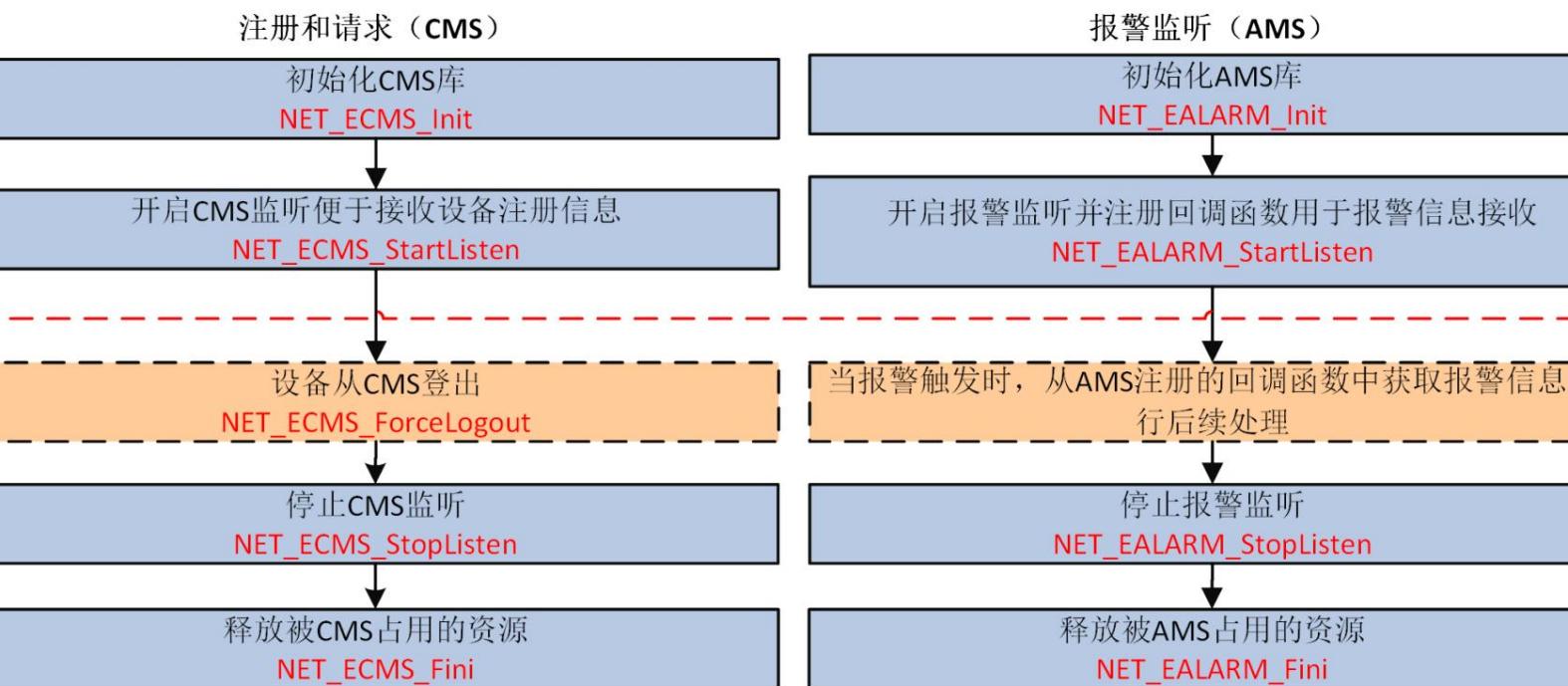


Figure 9-1 Flowchart of the interface call to enable the monitoring service to receive alarms

1. Call `NET_EALARM_StartListen` to start AMS alarm monitoring and register a callback function to receive alarm information.

The IP address and port number of the AMS are sent to the device by the CMS.

2. Optional operation: Set the SessionKey for the device that supports ISUP5.0.



SessionKey is obtained through the callback data type "enum_dev_sessionkey", see 3. When the alarm is triggered, the alarm information is obtained from the callback function registered by AMS for processing.

4. Call `NET_EALARM_StopListen` to stop AMS alarm monitoring. 5. Optional operation: Call

`NET_ECMS_ForceLogout` to force the device to log out.

Example:

Sample code to enable the monitoring service to receive alarms



In this sample code, CMS and AMS are installed in the same application on the same computer.

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

LONG ILoginID = -1;
LONG IListenHandle = -1;

/// ...

BOOL CALLBACK RegisterCallBack(LONG IUserID, DWORD dwDataType, void *Br, DWORD dwOutLen, void
nBr, DWORD dwInLen, void *pUser)
{
    if (ENUM_DEV_ON == dwDataType)
    {
        NET_EHOME_DEV_REG_INFO *pDevInfo = (NET_EHOME_DEV_REG_INFO *)Br

        if (pDevInfo != NULL)

        { ILoginID = IUserID;
            rntf(nn IUserID: %d, Device ID: %s\n", ILoginID, pDevInfo->byDeviceID);

        } //Input
        parameters NET_EHOME_SERVER_INFO *pServerInfo = (NET_EHOME_SERVER_INFO *)nBr
        pServerInfo->dwTimeOutCount = 6; //Heartbeat timeout times
        pServerInfo->dwKeepAliveSec = 15; //Heartbeat interval
        memcpy(pServerInfo->struUDPALarmSever.szIP, "10.16.2.123", sizeof("10.16.2.123")); //Alarm server IP address (TCP protocol)

pServerInfo->struUDPALarmSever.wPort = 7200; //Alarm server port (UDP protocol), which needs to be consistent with the port where the
alarm

        server starts listening pServerInfo->dwAlarmServerType = 0; //Alarm server type: 0-only supports UDP protocol reporting, 1-supports
both

        UDP and TCP protocol reporting} else if
(ENUM_DEV_OFF == dwDataType) {
            rntf(n IUserID: %d\n", IUserID);
            NET_ECMS_ForceLogout(IUserID);

        } else
        {

```

ISUPSDK (General) Development Guide

```

}

return TRUE;
}

///////////////////////////////
//AMS alarm callback function
BOOL CALLBACK AlarmMSGCallBack(LONG lHandle, NET_EHOME_ALARM_MSG *pAlarmMsg, void *pUserData) {

    lListenHandle = lHandle;

    DWORD dwType = pAlarmMsg->dwAlarmType; //Different alarm types (dwAlarmType), pAlarmInfo corresponds to different alarm information typesrntf(Cbc of
alarm
    listening, dwAlarmType[%d]\n", dwType);

    switch(dwType)

    { case EHOME_ALARM_UNKNOWN://Unknown alarm
        informationrntf(nnwn Alarm Type\n"); break;

    case EHOME_ALARM://Basic alarm information: motion detection, video cover, video loss, PIR alarm, face detection, area intrusion, etc.
        NET_EHOME_ALARM_INFO struAlarmInfo;
        memcpy(&struAlarmInfo, pAlarmMsg->pAlarmInfo, sizeof(NET_EHOME_ALARM_INFO)); rntf(Bc Alarm: Device ID[%s],
        szAlarmTime[%s], Alarm Type[%d], ctin VideoChannel[%d]\n", struAlarmInfo.szDeviceID, \ struAlarmInfo.szAlarmTime,
        struAlarmInfo.dwAlarmType,
        rrmnwrmctin
        struAlarmInfo.dwVideoChannel); break;

    case EHOME_ALARM_HEATMAP_REPORT://yyyyyy
        NET_EHOME_HEATMAP_REPORT struHeatmapInfo;
        memcpy(&struHeatmapInfo, pAlarmMsg->pAlarmInfo, sizeof(NET_EHOME_HEATMAP_REPORT)); rntf(m 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://Image recognition snapshot report upload
        NET_EHOME_FACESNAP_REPORT struFaceSnap;
        memcpy(&struFaceSnap, pAlarmMsg->pAlarmInfo, sizeof(NET_EHOME_FACESNAP_REPORT)); rntf(Fc 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 ȿȿȿ
        NET_EHOME_GPS_INFO struGpsInfo;
        memcpy(&struGpsInfo, pAlarmMsg->pAlarmInfo, sizeof(NET_EHOME_GPS_INFO)); rntf(G info: Device ID[%s],
        SampleTime[%s], Longitude[%d], tin struGpsInfo.byDeviceID, \ struGpsInfo.bySampleTime, struGpsInfo.dwLongitude, rGnwti ); break;

    case EHOME_ALARM_CID_REPORT://alarm host CID alarm information
        NET_EHOME_CID_INFO struCIDInfo;
        memcpy(&struCIDInfo, pAlarmMsg->pAlarmInfo, sizeof(NET_EHOME_CID_INFO));
}

```

```

rntf(C 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(){

/// ...

//AMS initialization
NET_EALARM_Init();

//Alarm listening
parameters NET_EHOME_ALARM_LISTEN_PARAM
struListen = {0}; memcpy(struListen.struAddress.szIP, "10.16.2.123",
sizeof("10.16.2.123")); struListen.struAddress.wPort = 7200; //Alarm
service listening port struListen.fnMsgCb = AlarmMSGCallBack; //
Alarm callback function
struListen.pUserData = NULL; struListen.byProtocolType = 1; //0- TCP mode (reserved, not supported yet), 1- UDP mode

//Start alarm monitoring
LONG lHandle = NET_EALARM_StartListen(&struListen);
if(lHandle < -1)
{
    rntf(NRMrn failed, error code: %d\n", NET_EALARM_GetLastError());
    NET_EALARM_Fini();
    return;
}

} rntf(NRMrn) //////////////////////

///////////////////////////////
//CMS registration module initialization
NET_ECMS_Init();

//Register listening
parameters 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;

//Start monitoring and receive device registration information. The registration callback function needs to send the alarm host IP and port to the device
LONG lListen = NET_ECMS_StartListen(&struCMSListenPara);
if(lListen < -1)

```

```

{
    rntf(NCMrn failed, error code: %d\n", NET_ECMS_GetLastError());
    NET_ECMS_Fini();
    return;

} rntf(NCMrn)

while(1 {

    Sleep(1000); //Perform subsequent operations after successful
    registration if(!LoginID
    >= 0) {
        break;
    }
}

Sleep(300000); //Wait for 5 minutes and receive alarm information
///////////////////////////////quit

//CMS stops listening
if(INET_ECMS_StopListen(lListen)) {

    rntf(NCMn failed, error code: %d\n", NET_ECMS_GetLastError());
}

//CMS deinitialization, release resources
NET_ECMS_Fini();

//AMS stops alarm monitoring
if(lListenHandle >= 0) {

    if (INET_EALARM_StopListen(lListenHandle)) {

        rntf(NRMn failed, error code: %d\n", NET_ECMS_GetLastError());
    }
}
//AMS deinitialization, release resources
NET_EALARM_Fini();

rntf(x)
}

```

Subsequent

processing • Call *NET_ECMS_StopListen* to stop CMS
 listening. • Call *NET_ECMS_Fini* and *NET_EALARM_Fini* to release the resources occupied by CMS and AMS.

Chapter 10 Equipment Maintenance

ISUPSDK provides interfaces and commands for obtaining device information (including version information) and device working status.

Restart the SDK, upgrade the device, or manually calibrate the time.

Get device information	Call <u>NCMGCn</u> and set dwCommand to "NET_EHOME_GET_DEVICE_CFG" (Command No.: 3). Device information is returned by pOutBuf through the structure <u>NET_EHOME_DEVICE_CFG</u> times.
Configuring device information	Call <u>NCMGCn</u> and set dwCommand to "NET_EHOME_SET_DEVICE_CFG" (Command No.: 4), and set pInBuf to <u>NET_EHOME_DEVICE_CFG</u> .
Query the working status of the device	<u>GETDEVICEWORKSTATUS</u>
Reboot the device	<u>REBOOT</u>
Upgrading Equipment	<u>UPDATE</u>
Performing manual time calibration	<u>ADJUSTTIME</u>
Get device version information	Call <u>NCMGCn</u> and set dwCommand to "NET_EHOME_GET_VERSION_INFO" (Command No. 2). Device version information is provided by pOutBuf through the structure <u>NET_EHOME_DEVICE_CFG</u> is returned.

Chapter 11 Storage Management

The storage management function integrated through ISUP SDK relies on the storage management service (SS). SS supports saving files and supports multiple protocol types, such as Tomcat protocol, VRB protocol, and ISUP5.0.

The following two diagrams show the role of SS in file upload and download applications.

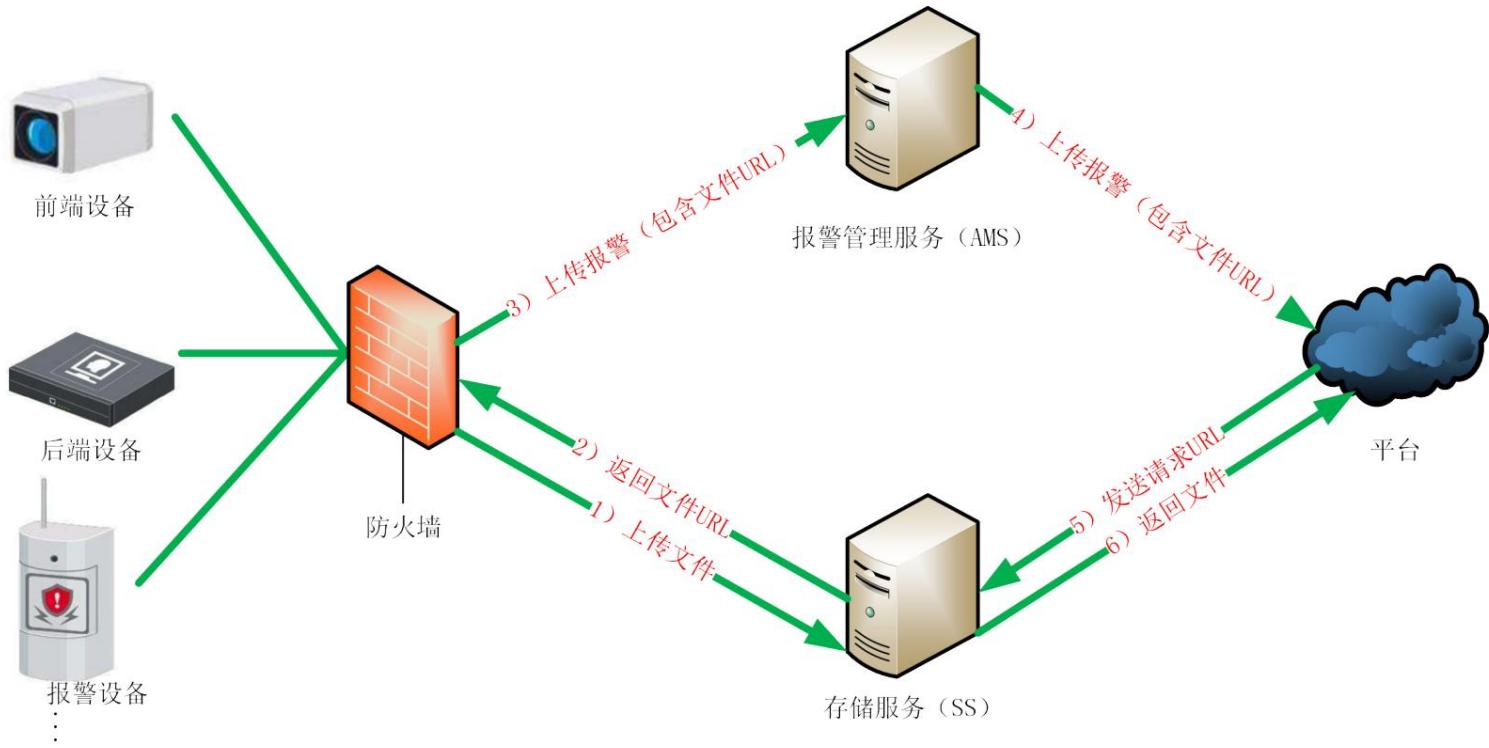


Figure 11-1 Device uploads files to the platform

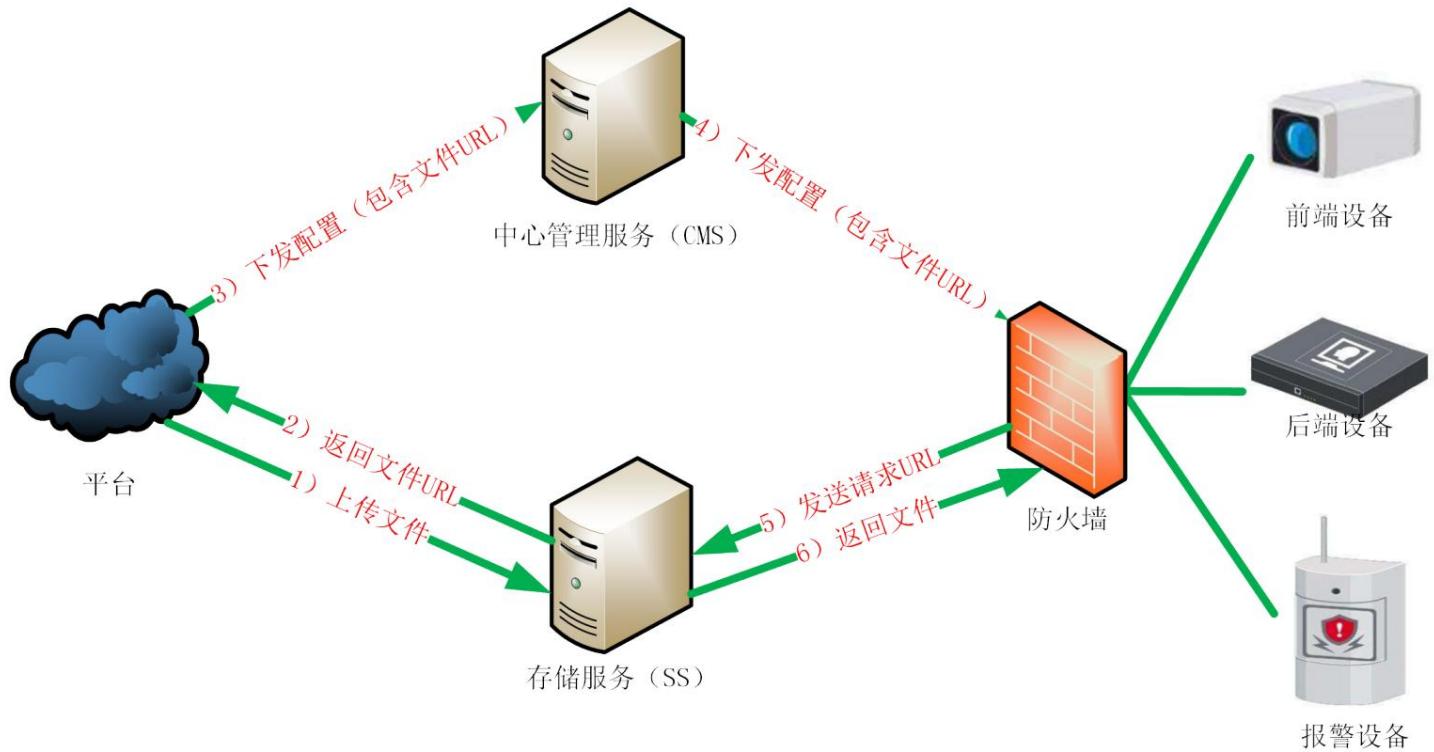


Figure 11-2 Platform sends files to device

11.1 Enable storage service

Before you start uploading files to or downloading files from the storage service (SS), you must enable the storage service. In addition, you can enable the safe mode.

Steps

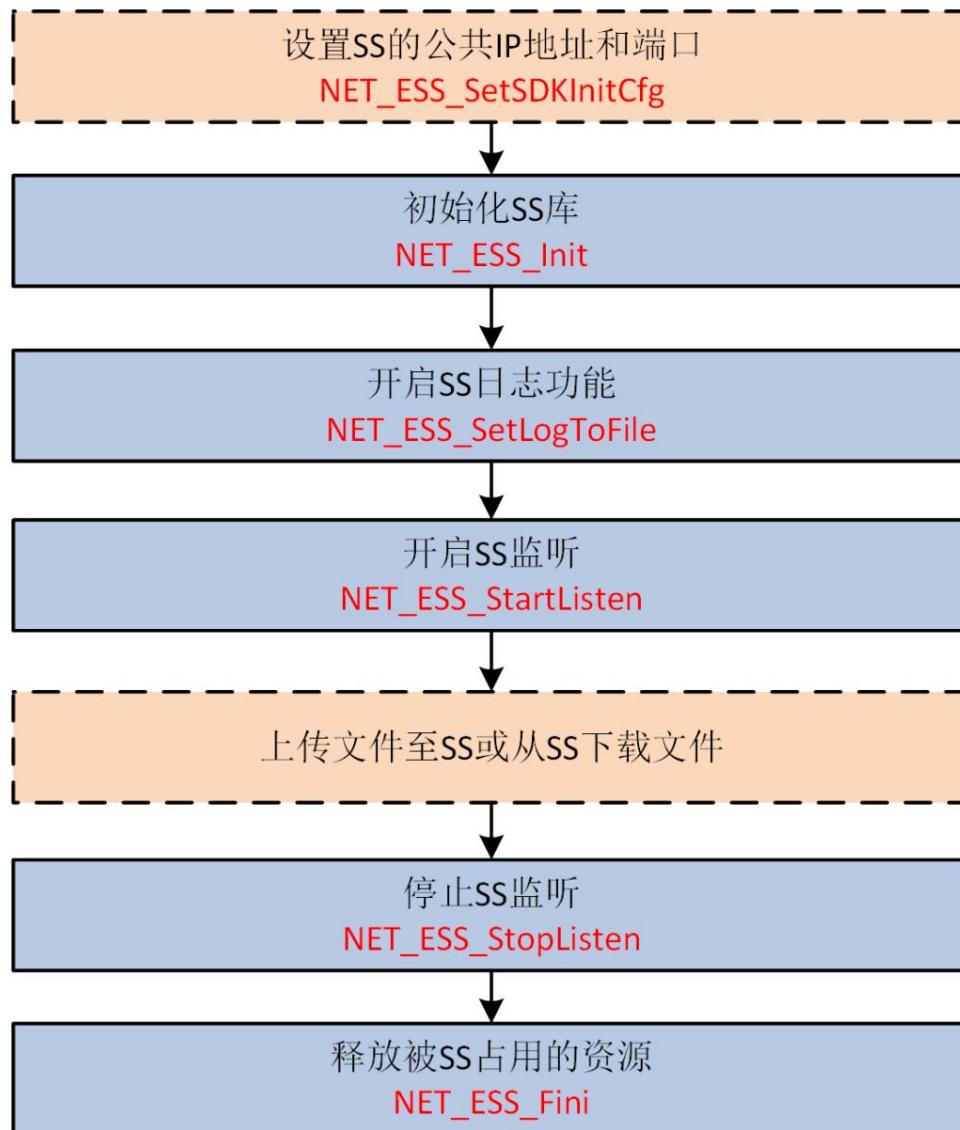


Figure 11-3 Flowchart of the interface call to enable the storage service

1. Optional operation: Call `NET_ESS_SetSDKInitCfg` and set `enumType` to

"`NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT`" and set `nB` to `NET_EHOME_IPADDRESS` to set the public IP address and port of SS.



When SS is in a public network environment, this step is

mandatory. 2. Call `NET_ESS_Init` to

initialize SS. 3. Call `NET_ESS_SetLogFile` to enable the log

function for SS. 4. Call `NET_ESS_StartListen` to enable SS listening.

ISUPSDK (General) Development Guide



This interface can be used to enable the security mode. In this mode, files must be authenticated before they can be uploaded, downloaded, or deleted. **5. Optional operations:**

Upload files to SS or download files from SS. **6. Call NET_ESS_StopListen to stop SS**

listening. _____

Example

code for enabling storage service

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

n SS_STORAGE_PATH "C:\\Picture" //File save path n
PIC_URI_LEN 128 //Picture URL size

//Message callback
function int SS_Message_Callback(LONG iHandle, NET_EHOME_SS_MSG_TYPE enumType
. void Br DWORD dwOutLen, void nBr DWORD dwInLen, void *pUser)
{
    if (enumType == NET_EHOME_SS_MSG_TOMCAT) ////
        //Callback Tomcat protocol information

        LPNET_EHOME_SS_TOMCAT_MSG pTomcatMsg = (NMMCMSG)Br char szPicUri[PIC_URI_LEN] for (int i = 0; i <
        pTomcatMsg->dwPicNum; i++)    * 4] = { 0 };
    {

        rntf(cr + 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 };
    rntf(M "tomcat:uri[%s],picNum[%d] ,picInfo[%s]" , szUrlHead,
    picNum, szPicUri); rntf(M)

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)
```

```
{  
    (B)nBr = 1;  
  
} else if (enumType == NET_EHOME_SS_MSG_CLOUD_AK) {  
  
    rnc((cr)nBr 5998bb44c783099c4073V7F dwInLen);  
  
    } return TRUE;  
}  
  
//Callback function for saving files  
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;  
}  
  
//Judge whether the file storage path already exists. If not, create a file if (!  
PathIsDirectory(SS_STORAGE_PATH)) {  
  
    if (!CreateDirectory(SS_STORAGE_PATH, NULL)) {  
  
        return FALSE;  
    }  
}  
  
//Save the file  
char strFilePath[MAX_PATH] = { 0 };  
mft(rF "%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() {
```

```

// Initialize resources
NET_ESS_Init(); // Open the log function
NET_ESS_SetLogToFile(3, "C:/SdkLog/", TRUE);

//yyyy
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.fnSSorageCb = SS_Storage_Callback;
struSSListenParam.pUserData = NULL; long
m_ISSHandle = NET_ESS_StartListen(&struSSListenParam); if (-1 == m_ISSHandle) {

    rntf(Nrn Failed, porty%d", struSSListenParam.struAddress.wPort);

} else
{
    rntf(Nrn succ porty%d", struSSListenParam.struAddress.wPort);
}

char cTmp = '0'; do
{

    rntf(n q to exit!"); cTmp =
    getchar(); }while(cTmp !
= 'q');

//Stop monitoring
NET_ESS_StopListen(m_ISSHandle); //Release
resources
NET_ESS_Finished();

return 0;
}

```

Subsequent

processing calls NET_ESS_Fini to release the resources occupied by SS.

11.2 Uploading files to storage services

During the ISUPSDK integration process, the file format transmitted between the platform and the device must be a URL, and the file information can be downloaded through the URL. Therefore, before uploading files to the platform or sending files to the device or platform must upload the files to the storage service (SS) to obtain the file URL. In addition, it supports uploading or deleting files in safe mode.

Prerequisites:

Make sure SS is enabled. [Enable storage service](#)

Steps

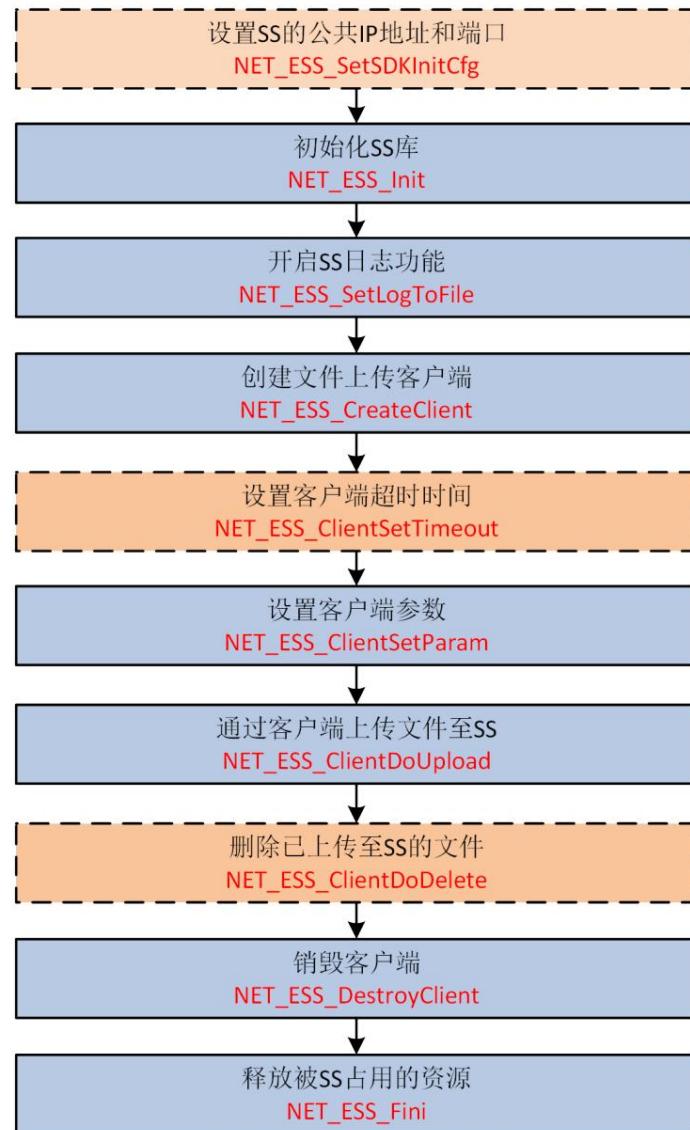


Figure 11-4 API call flow chart for uploading files to the storage service

ISUPSDK (General) Development Guide

1. Optional operation: Call NET_ESS_SetSDKInitCfg and set enumType to

"NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT" and set nB to NET_EHOME_IPADDRESS to set the public IP address and port of SS.



When SS is in a public network environment, this step is mandatory. **2.**

Call NET_ESS_Init to initialize SS. **3.** Call

NET_ESS_SetLogFile to enable the log function for SS. **4.** Call

NET_ESS_CreateClient to create a file upload client.



In safe mode, the client type cannot be "NET_EHOME_SS_CLIENT_TYPE_TOMCAT" (client supporting Tomcat protocol) and "NET_EHOME_SS_CLIENT_TYPE_VRB" (client supporting VRB protocol).

5. Optional operation: Call NET_ESS_ClientSetTimeout to set the upload client timeout. **6.** Call NET_ESS_ClientSetParam

to set the upload client parameters.



In security mode, file authentication methods include the Key Management Service (KMS) and cloud storage services, which means you need to set a user name and password for the KMS, or set an AccessKey and SecretKey for the cloud storage service.

7. Call NET_ESS_ClientDoUpload to upload the file to SS through the client.

The uploaded file will be saved in SS, and the corresponding file URL will be returned to the device.

8. Optional operation: Call NET_ESS_ClientToDelete to delete the files uploaded to SS. **9.** After the upload

is completed, call NET_ESS_DestroyClient to destroy the upload client.

Example (non-secure mode) Sample code for uploading files to the storage service

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

int main()

{ // Initialize SS
    NET_ESS_Init(); //
    Open the log function
    NET_ESS_SetLogFile(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;
```

ISUPSDK (General) Development Guide

```

//Upload files to SS based on Tomcat protocol
ISSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (ISSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(ISSClientHandle, 60 * 1000);           * 1000, 60
    NET_ESS_ClientSetParam(ISSClientHandle, 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(ISSClientHandle);
}

//Upload files to SS based on VRB protocol
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_VRB;
ISSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (ISSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(ISSClientHandle, 60           * 1000, 60 * 1000);

    NET_ESS_ClientSetParam(ISSClientHandle, SS_CLIENT_FILE_PATH_PARAM_NAME, "C:\Picture\1.jpg");
    NET_ESS_ClientSetParam(ISSClientHandle, SS_CLIENT_VRB_FILENAME_CODE, nm1c)

    NET_ESS_ClientDoUpload(ISSClientHandle, szUrl, MAX_URL_LEN_SS - 1);

    NET_ESS_DestroyClient(ISSClientHandle);
}

//Upload files to SS based on KMS protocol
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_KMS;
ISSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (ISSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(ISSClientHandle, 60           * 1000, 60 * 1000);

    NET_ESS_ClientSetParam(ISSClientHandle, 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(ISSClientHandle);
}

//Upload files to SS based on ISUP5.0
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_CLOUD;
ISSClientHandle = NET_ESS_CreateClient(&struClientParam);
if (ISSClientHandle >= 0)
{
    NET_ESS_ClientSetTimeout(ISSClientHandle, 60           * 1000, 60 * 1000);
}

```

```
NET_ESS_ClientSetParam(ISSClientHandle, SS_CLIENT_FILE_PATH_PARAM_NAME, "C:\\Picture\\1.jpg"); NET_ESS_ClientSetParam(ISSClientHandle,
SS_CLIENT_CLOUD_AK_NAME, "test"); NET_ESS_ClientSetParam(ISSClientHandle, 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(ISSClientHandle);

}

//Release the resources occupied by SS
NET_ESS_Finished();

return 0;
}
```

Subsequent

processing calls *NET_ESS_Fini* to release the resources occupied by SS.

11.3 Downloading files from storage services

During the ISUPSDK integration process, since the file is transferred in URL format and the file details are saved in the storage service, users need to download the file from the storage service according to the file URL. In addition, file downloading in safe mode is supported.

Prerequisites:

Make sure SS is enabled.

[Enable storage service](#) 9

Steps

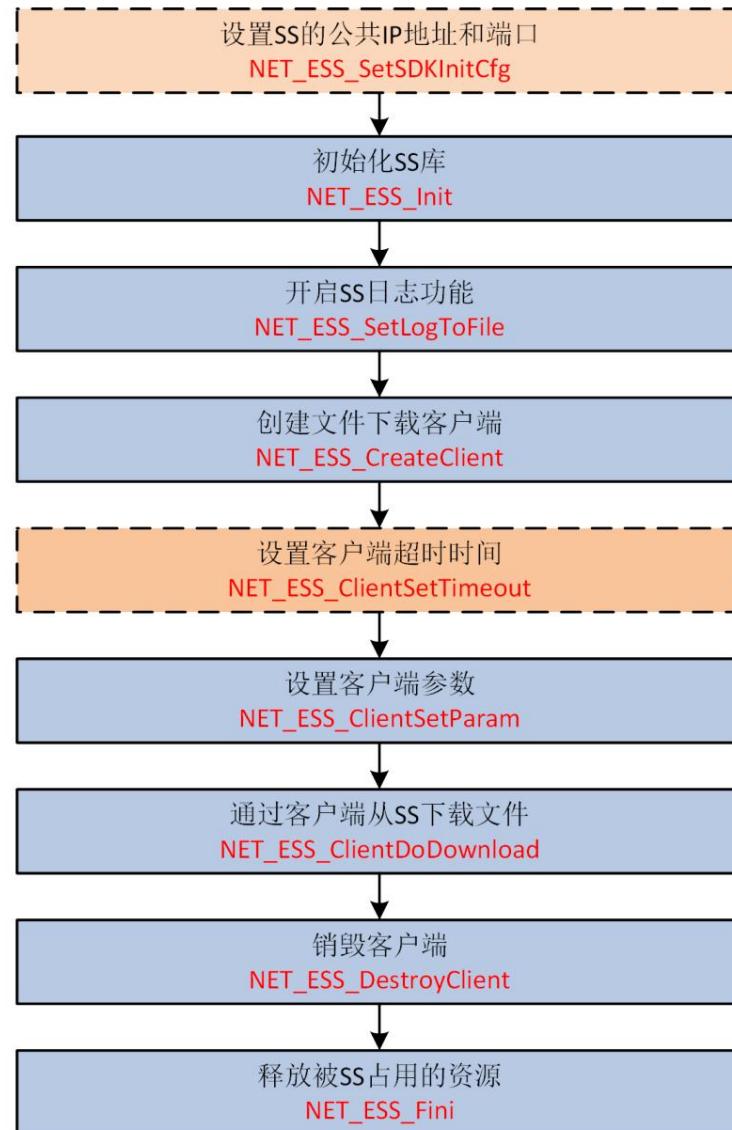


Figure 11-5 API call flow chart for downloading files from storage service

1. Optional operation: Call `NET_ESS_SetSDKInitCfg` and set `enumType` to

"`NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT`" and set `nB` to `NET_EHOME_IPADDRESS` to set the public IP address and port of SS.



When SS is in a public network environment, this step is

mandatory. 2. Call `NET_ESS_Init` to

initialize SS. 3. Call `NET_ESS_SetLogFile` to enable the log

function for SS. 4. Call `NET_ESS_CreateClient` to create a file download client.

ISUPSDK (General) Development Guide



In safe mode, the client type cannot be "NET_EHOME_SS_CLIENT_TYPE_TOMCAT" (client supporting Tomcat protocol) and "NET_EHOME_SS_CLIENT_TYPE_VRB" (client supporting VRB protocol).

5. Optional operation: Call `NET_ESS_ClientSetTimeout` to set the timeout period for downloading the client. 6. Call `NET_ESS_ClientSetParam` to set the parameters for downloading the client.



In security mode, file authentication methods include the Key Management Service (KMS) and cloud storage services, which means you need to set a user name and password for the KMS, or set an AccessKey and SecretKey for the cloud storage service.

7. Call `NET_ESS_ClientDoDownload` to download the file from SS through the download client according to the file URL. 8. After the download is completed, call `NET_ESS_DestroyClient` to destroy the download client.

Example

(non-secure mode) Sample code for downloading a file from a storage server

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

int main()

{ // Initialize SS
    NET_ESS_Init(); //
    Open the log function
    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;

    //Download files from SS based on Tomcat protocol and VRB
    protocol struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_VRB;
    ISSClientHandle = NET_ESS_CreateClient(&struClientParam); if
    (ISSClientHandle >= 0) {

        NET_ESS_ClientSetTimeout(ISSClientHandle, 60 * 1000, 60 * 1000);

        byte* bFileContent = NULL;
        DWORD dwFileLen = 0; if
        (NET_ESS_ClientDoDownload(ISSClientHandle, szUrl, (void**)&bFileContent, dwFileLen)) {

            FILE* pFile = fopen("C://Picture//vrb.jpg", "wb+"); if (pFile !=
            NULL) {

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

ISUPSDK (General) Development Guide

```

    }

}

//Destroy the download client
NET_ESS_DestroyClient(ISSClientHandle);
}

//Download files from SS based on KMS protocol
ISSClientHandle = NET_ESS_CreateClient(&struClientParam); if (ISSClientHandle >= 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(ISSClientHandle, 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();

    //Destroy the download client
    NET_ESS_DestroyClient(ISSClientHandle);
}

//Based on ISUP version 5.0, download files from SS
struClientParam.enumType = NET_EHOME_SS_CLIENT_TYPE_CLOUD; ISSClientHandle =
NET_ESS_CreateClient(&struClientParam); if (ISSClientHandle >= 0) {

    NET_ESS_ClientSetTimeout(ISSClientHandle, 60 * 1000);

    NET_ESS_ClientSetParam(ISSClientHandle, SS_CLIENT_CLOUD_AK_NAME, "test");
    NET_ESS_ClientSetParam(ISSClientHandle, SS_CLIENT_CLOUD_SK_NAME, "12345");

    byte* bFileContent = NULL;
    DWORD dwFileLen = 0; if
    (NET_ESS_ClientDoDownload(ISSClientHandle, szUrl, (void**)&bFileContent, dwFileLen)) {

        FILE* pFile = fopen("C://Picture//cloud.jpg", "wb+"); if (pFile != NULL) {

            fwrite(bFileContent, 1, dwFileLen, pFile);

```

ISUPSDK (General) Development Guide

```
fclose(pFile); }

}

NET_ESS_DestroyClient(ISSClientHandle);

}

//Release the resources occupied by SS
NET_ESS_Finished();

return 0;
}
```

Subsequent

processing calls NET_ESS_Fini to release the resources occupied by SS.

Chapter 12 Parameter Configuration

This chapter shows the basic or advanced parameter configurations supported by ISUPSDK, such as network, encoding, image, video and serial port.

Table 12-1 Network configuration

Get network parameters	<p>Call <code>NCMGCn</code> and set <code>dwCommand</code> to "NET_EHOME_GET_NETWORK_CFG" (command number: 5).</p> <p>The network parameters are returned by <code>pOutBuf</code> through the structure <code>NET_EHOME_NETWORK_CFG</code>.</p>
Configure network parameters	<p>Call <code>NCMGCn</code>, set <code>dwCommand</code> to "NET_EHOME_SET_NETWORK_CFG" (command number: 6), and set <code>pInBuf</code> to <code>NET_EHOME_NETWORK_CFG</code>.</p>

Table 12-2 Encoding Configuration

Get encoding parameters	<p>Call <code>NCMGCn</code> with <code>dwCommand</code> set to "NET_EHOME_GET_COMPRESSION_CFG" (command number: 7) and <code>pCondBuf</code> set to <code>NET_EHOME_COMPRESSION_COND</code>.</p> <p>The encoding parameters are returned by <code>pOutBuf</code> through the structure <code>NET_EHOME_COMPRESSION_CFG</code>.</p>
Configure encoding parameters	<p>Call <code>NCMGCn</code> with <code>dwCommand</code> set to "NET_EHOME_SET_COMPRESSION_CFG" (command number: 8), <code>pCondBuf</code> set to <code>NET_EHOME_COMPRESSION_COND</code>, and <code>pInBuf</code> set to <code>NET_EHOME_NETWORK_CFG</code>.</p>

Table 12-3 Image and OSD configuration

Get OSD parameters	Call <code>NCMGCn</code> and set <code>dwCommand</code> to "NET_EHOME_GET_PIC_CFG" (command number: 23).
--------------------	--

ISUPSDK (General) Development Guide

	The OSD parameters are returned by pOutBuf through the structure <u>NET_EHOME_PIC_CFG</u> .
Configuring OSD parameters	Call <u>NCMCn</u> and set dwCommand to "NET_EHOME_SET_PIC_CFG" (Command No.: 24), and set pInBuf to <u>NET_EHOME_PIC_CFG</u> \ddagger
Get image parameters	Call <u>NCMGCn</u> and set dwCommand to "NET_EHOME_GET_IMAGE_CFG" (Command No.: 9), and set pCondBuf to It is a 4-digit channel number. Image parameters are returned by pOutBuf through the structure <u>NET_EHOME_IMAGE_CFG</u> times.
Configure image parameters	Call <u>NCMCn</u> and set dwCommand to "NET_EHOME_SET_IMAGE_CFG" (command number: 10), set pCondBuf to 4-bit channel number and set pInBuf to <u>NET_EHOME_IMAGE_CFG</u> \ddagger

Table 12-4 Video recording configuration

Get recording parameters	Call <u>NCMMCn</u> to transmit the command message <u>GetRecordPlanPara</u> \ddagger
Set recording parameters	Call <u>NCMMCn</u> to transmit the command message <u>SetRecordPlanPara</u> \ddagger

Table 12-5 Serial port configuration

Get RS-485 serial port parameters	Call <u>NCMMCn</u> to transmit the command message <u>GetRS485Para</u> .
Configure RS-485 serial port parameters	Call <u>NCMMCn</u> to transmit the command message <u>SetRS485Para</u> .
Get RS-232 serial port parameters	Call <u>NCMMCn</u> to transmit the command message <u>GetRS232Para</u> .
Configure RS-232 serial port parameters	Call <u>NCMMCn</u> to transmit the command message <u>SetRS232Para</u> .

Table 12-6 Other configurations

Remote arming and disarming	Call <code>NET_ECMS_XMLRemoteControl</code> to transmit command message <code>SETDEFENSESTATUS</code>
Restart Algorithm Library	Call <code>NET_ECMS_XMLRemoteControl</code> to transmit the command message <code>RESTARTIVSLIB</code> .

12.1 Configuring Power Mode

After turning on the device's sleep and wake-up mode, the device can be used longer with the same amount of power. When the device is in sleep mode, any signaling interface such as preview, playback, voice intercom, configuration, etc. is invalid. Only after waking up the device can normal signaling interaction be restored.

Prerequisites •

Make sure the device is online using 4G/Wi-Fi to use low power consumption. • Make sure the device has been registered on the Central Management Service (CMS), see [Device Registration](#) to V2.3.5.1 or higher.

Steps



ISUPSDK versions lower than V2.3.5.1 do not support the sleep-wakeup interface, and there are compatibility issues after connecting to the sleep-wakeup device.

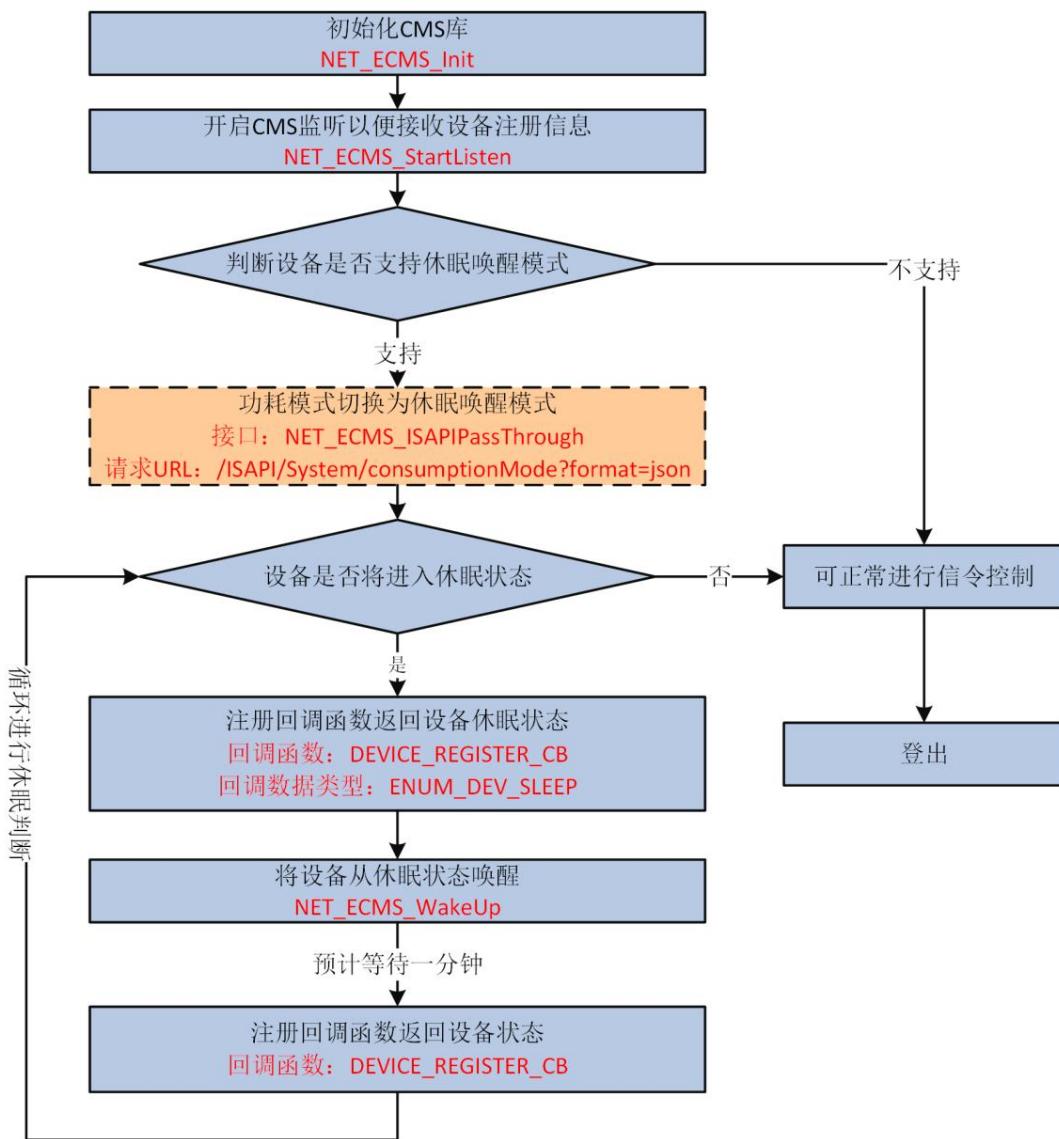


Figure 12-1 Interface call flow chart for configuring power

consumption mode 1. Call `NET_ECMS_ISAPIPassThrough` transmission request URI: `GET ymcb` to obtain the device capability set to determine whether the device supports power consumption mode configuration.

The device capability set `XML_DeviceCap` is returned by B through the structure `NET_EHOME_PTXML_PARAM`.

If the node `<CnmnM>` is returned in the message and its value is true, it means that the device supports power consumption mode configuration and you can proceed to the subsequent steps.

Otherwise, it means that the device does not support power consumption mode configuration. Please stop executing this task.

2. Call `NET_ECMS_ISAPIPassThrough` to transfer request URI: `GET ymcnmnM`

`cbmn` obtains the power consumption mode configuration capability to determine whether the device supports the sleep wake-up mode.

ISUPSDK (General) Development Guide

Power mode configuration capability message *NCCnmnM* through the structure

NET_EHOME_PTXML_PARAM is returned by **B**.

If the **devWorkMode** node in the message supports "sleepOrWakeUp", it means that the device supports the sleep-wakeup mode.

3. Optional operation: Call *NET_ECMS_ISAPIPassThrough* to transfer the request URI: PUT *ym*

cnmnMmn, and set the value of **devWorkMode** in the request message *NCnmnM* to "sleepOrWakeUp".



illustrate

The default power consumption mode of the device is sleep-wake mode. If the sleep-wake mode is already enabled, you can ignore this step.

The device's power consumption mode switches to sleep wake-up

mode. 4. The device sleep state will be called back to the registered callback function through the callback data type "ENUM_DEV_SLEEP".

DEVICE_REGISTER_CB



illustrate

If no operation is performed on the device within 5 minutes, or the device power is lower than the set threshold, the device will enter sleep mode.

state.

• The device will not respond to any signaling control (except the sleep wake-up interface) when it is in sleep mode, and the signaling interface control will return the error code 8301—"NET_ERR_DEV_SLEEP". 5. Call

NET_ECMS_WakeUp to wake the device from sleep mode.



illustrate

• Different devices require different wake-up times, generally estimated to be 1 minute. • Some devices (such as devices connected via Wi-Fi) have a heartbeat timeout of 90 seconds. If the wake-up time exceeds this time, the device will automatically go offline and call back the offline status. The user needs to re-register the device.

The device will be powered on again and registered, and the device status will be called back to the registration callback function *DEVICE_REGISTER_CB*.

Subsequent

processing • Call *NET_ECMS_StopListen* to stop the CMS listening service. • Call

NET_ECMS_Fini to release the resources occupied by CMS.

Chapter 13 Integration via the Transfer Text Protocol

ISUPSDK supports integration through the transmission text protocol. The text protocol content includes the request method, request URI, query parameters, request and response messages. Application integration can be achieved without any other operations between the device and the platform (or system).

Prerequisites

Make sure that `NET_ECMS_Init` has been called to initialize CMS.

Steps

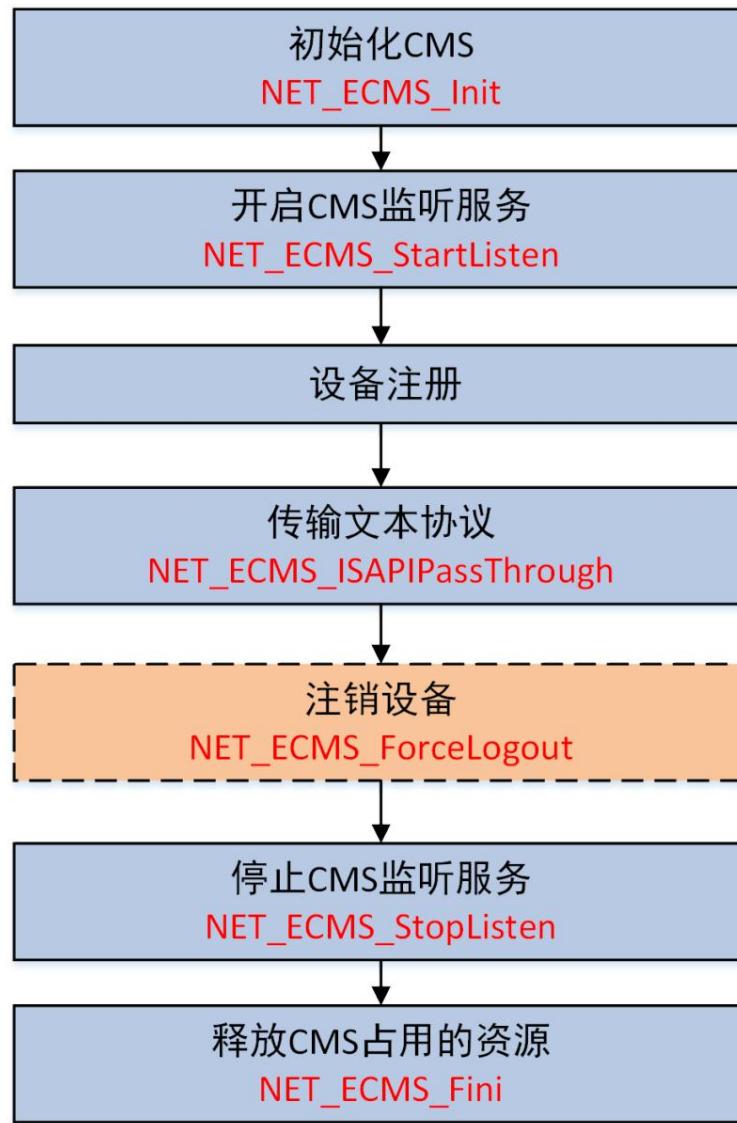


Figure 13-1 Interface call flow for integration through the text transmission protocol

ISUPSDK (General) Development Guide

1. Call NET_ECMS_StartListen to start the CMS listening service to receive device registration information.



illustrate

The device registration information will be viewed in the structure NET_EHOME_DEV_REG_INFO returned by the registration callback function
(DEVICE_REGISTER_CB).

2. Register device information.



For

information about the message processing flow for device registration, Device registration message processing flow

3. Call NET_ECMS_ISAPIPassThrough to transfer the text protocol, including the request method, request URI, query parameters,

Sum the response message and implement the corresponding

application. 4. Optional operation: Call NET_ECMS_ForceLogout to force the device to log out.

Subsequent

processing • Call NET_ECMS_StopListen to stop the CMS listening service. • Call NET_ECMS_Fini

to release the resources occupied by CMS.

Chapter 14 Interface Reference

14.1 Registration Module

14.1.1 Basic Interface

NET_ECMS_Init

Initialize the registration module library.

Interface definition

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

Return value

TRUE indicates success, *FALSE* indicates failure.

Remark

- If initialization fails, please check the path of the loaded library file and the system environment. •

This interface must be called in conjunction with *NET_ECMS_Fini*. •

The *NET_ECMS_SetSDKInitCfg* interface can be called before calling this interface, and other interfaces must be called after this interface. use.

NET_ECMS_Fini

Deinitializes the registry and releases resources used by the Central Management Server (CMS).

Interface definition

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

Return value

TRUE indicates success, *FALSE* indicates failure.

Note:

This interface must be called together with *NET_ECMS_Init*.

NET_ECMS_ForceLogout

The device is forcibly logged out and the device status changes to offline.

Interface definition

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

parameter

lUserID

[IN] User ID, returned by the [callback function configured by NET_ECMS_StartListen](#).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call

[NET_ECMS_GetLastError](#) to get the error code.

Notes •

If the device is forcibly deregistered by calling this interface (the device status becomes offline), CMS will no longer respond to any request with the device user ID.

Once the device status becomes online, the registration request will be called back to the user, who will decide whether the device has been registered. • When CMS receives a callback

message that the device is offline, it is recommended to call this interface to release resources.

NET_ECMS_SetSDKInitCfg

Sets the Central Management Server (CMS) initialization parameters.

Interface definition

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

parameter

enumType

[IN] Initialization configuration type, see enumeration [NET_EHOME_CMS_INIT_CFG_TYPE](#) for details.

nB

[IN] Initialize configuration parameters, depending on the configuration type (enumType), see the corresponding relationship below.

enumType	nB
NET_EHOME_CMS_INIT_CFG_LIBEAY_PATH	Path with the library names of the OpenSSL cryptographic libraries.
NET_EHOME_CMS_INIT_CFG_SSLEAY_PATH	Path with the library name of the OpenSSL communication library.

Return value

TRUE indicates success, *FALSE* indicates failure.

Note :

This interface must be called before initializing *NET_ECMS_Init*, otherwise the SDK cannot guarantee that the settings will take effect. • If the call fails, the error code cannot be obtained or the correct error code cannot be obtained, because the error code must be initialized before it can be obtained.

Obtain.

- This interface does not check the validity of the OpenSSL path. The path is verified when the library is loaded. If the path is incorrect, other

The relevant interfaces will not report errors, but errors will be written to the log.

- When multiple modules are used in the same process, it is recommended to set the same OpenSSL path for all modules. For example, if the Registration Module (CMS) and the Alarm Management Module (AMS) are used in the same process, if the CMS has OpenSSL loaded, but the OpenSSL path of AMS is different from that of CMS, and the version number of the library is also different, AMS will use the loaded OpenSSL.

- Different modules (AMS/CMS/SS/SMS) are located in the same process. When loading the OpenSSL library, ensure that different modules use

The OpenSSL library path used is valid, and the OpenSSL library version and library name are the same.

NET_ECMS_GetBuildVersion

Gets the version information of the Central Management Server (CMS).

Interface definition

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

The return

value returns the version number. The upper two bytes represent the major version, and the lower two bytes represent the minor version, for example: 0x02040001 (version 2.4.0.1).

NET_ECMS_SetLogFile

Sets parameters to enable the Central Management Server (CMS) logging function.

Interface definition

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

parameter

nLogLevel

[IN] Log type: 0-Enable logging (default), 1-Error log, 2-Error and debug log, 3-Error, debug and information log.

strLogDir

[IN] Log file save path. The default save path in Windows operating system is "C:\\SdkLog\\", and the default save path in Linux operating system is "/home/sdklog/". If you want to use the default save path, set this parameter to "NULL".

bAutoDel

[IN] Whether to automatically delete the log file after a certain period of time: "TRUE" - yes (default), "FALSE" - no.

When "FALSE", log compression will be enabled. Every time 50 log files are generated, log compression will be performed (the 50 generated log files will be compressed into one compressed file).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call *NET_ECMS_GetLastError* to get the error code.

Notes

- The path to save the log file must be an absolute path and must end with "\\", for example: "C:\\SdkLog\\". • If the save path is changed, the changed path will be used to save the next log file. • When the value of **bAutoDel** is "TRUE", the first log file will not be deleted because the file contains the start time.
 - Allow HCISUPSDK to modify log parameters during runtime. You can add the HCISUPSDK_Log_Switch.xml file in the runtime directory to configure the runtime log parameters.

[Log parameter configuration](#)

NET_ECMS_GetSDKLocalCfg

Gets the local configuration parameters of the Central Management Server (CMS).

Interface definition

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

parameter

enumType

[IN] Supported local configuration type macro definition values: -1, 0, 1, 2, 3, 4, 5 and 6, see enumeration definition

[NET_EHOME_LOCAL_CFG_TYPE](#)

B

[OUT] Local configuration parameters, related to the local configuration type (enumType), see the table below for details.

enumType	B	illustrate
UNDEFINE	Undefined.	Undefined.
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u>	Get CMS module device active Access security information.
AMS_ADDRESS	<u>NET_EHOME_AMS_ADDRESS</u>	Get the monitoring address of CMS alarm site.
SEND_PARAM	<u>NET_EHOME_SEND_PARAM</u>	Get the data of CMS module Send parameters (such as timeout, etc.).
COM_PATH	COM path, pointing to a string buffer. Get the CMS module	HCAapSDKCom (for HCISUP related dependency libraries path to the directory.
SESSIONKEY_REQ_MOD	points to a BOOL variable.	Get whether to proceed SessionKey request callback (only For ISUP5.0 protocol true means that ISUP5.0 device registration callback ENUM_DEV_

enumType	B	illustrate
		SESSIONKEY_REQ callback; false means in ISUP5.0 Not generated in the standby registration callback ENUM_DEV_ SESSIONKEY_REQ callback. Default is false.
DEV_DAS_PINGREO_CALLBACK	<i>NET_EHOME_LOCAL_DEV_PINGREO</i> gets whether to perform	ENUM_DEV_DAS_PINGREO callback in the registration callback. It is only for ISUP5.0 protocol devices. By default, no callback is performed.
REGISTER_LISTEN_MODE	<i>NET_EHOME_REGISTER_LISTEN_MODE</i> gets the registration listening mode of the CMS module.	

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the *error code*.

See also

[*NET_ECMS_SetSDKLocalCfg*](#)

NET_ECMS_SetSDKLocalCfg

Set local configuration parameters for the Central Management Server (CMS).

Interface definition

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

parameter

enumType

[IN] Supported local configuration type macro definition values: 0, 1, 2, 3, 4, 5 and 6, see enumeration definition

NET EHOME LOCAL CFG TYPE y

nB

[IN] Local configuration parameters, related to the local configuration type (enumType), see the table below for details.

enumType	B	illustrate
UNDEFINE	Undefined	Undefined.
ACTIVE_ACCESS_SECURITY	<u>module device's active NET_EHOME_LOCAL_ACCESS_SECURITY</u> ÿ	Set the security information of the CMS access.
AMS_ADDRESS	<u>NET_EHOME_AMS_ADDRESS</u> ÿ	Set the monitoring address for CMS alarms.
SEND_PARAM	Set network data sending parameters, the corresponding structure is <u>NET_EHOME_SEND_PARAM</u> .	Set the data sending parameters of the CMS module (such as timeout, etc.).
SET_REREGISTER_MODE	sets the repeated registration mode of the 2.0 protocol device. The corresponding structure is <u>NET_EHOME_SET_REREGISTER_MODE</u> ÿ	Set the device re-registration mode, only for ISUP2.0 protocol devices.
COM_PATH	COM path, pointing to a string buffer (for example: "./HCAapSDKCom/").	Set the HCAapSDKCom of the CMS module (for Path to the directory of HCISUP related dependency libraries).
SESSIONKEY_REQ_MOD	points to a BOOL variable.	Set whether to perform SessionKey request callback (only for ISUP5.0 protocol devices). true means in the ISUP5.0 device registration callback ENUM_DEV_SESSIONKEY_REQ callback; false means in ISUP5.0

enumType	B	illustrate
		Not performed in the backup registration callback ENUM_DEV_ SESSIONKEY_REQ callback. Default is false.
DEV_DAS_PINGREO_CALLBACK	<i>NET_EHOME_LOCAL_DEV_PINGREO</i> sets whether to perform ENUM_DEV_DAS_PINGREO callback in the registration callback. It is only for ISUP5.0 protocol devices and no callback is performed by default.	
REGISTER_LISTEN_MODE	<i>NET_EHOME_REGISTER_LISTEN_MODE</i> sets the registration listening mode of the CMS module.	

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned,

call *NET_ECMS_GetLastError* to get the *error code*.

See also

[*NET_ECMS_GetSDKLocalCfg*](#)

NET_ECMS_GetLastError

Gets the error code if the call fails or completes.

Interface definition

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

Return value

returns error code, see [ISUPSDK for details](#) *error code*

Note:

This interface needs to be called after *NET_ECMS_Init*.

NET_ECMS_WakeUp

Wake the device from sleep

Interface definition

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

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen*.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

14.1.2 Configuration and Control Interface**NET_ECMS_GetDeviceSessionKey**

Get session key information for devices that support ISUP version 5.0.

Interface definition

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

parameter

deviceKey

[OUT] Device session key information supporting ISUP version 5.0. For details, see the structure

NET_EHOME_DEV_SESSIONKEY

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

See also

[NET_ECMS_SetDeviceSessionKey](#)

NET_ECMS_SetDeviceSessionKey

Set the session key for devices that support ISUP version 5.0.

Interface definition

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

parameter

deviceKey

[IN] Device session key information supporting ISUP version 5.0. For details, see the structure

[NET_EHOME_DEV_SESSIONKEY](#)

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned,

call [NET_ECMS_GetLastError](#) to get the error code.

Notes •

Generally, this interface is called when registering the callback ENUM_DEV_SESSIONKEY so that the 5.0 protocol device can complete the

registration. • When there is no load balancing (LBS), this interface is generally called when the CMS registers the callback ENUM_DEV_SESSIONKEY. •

When there is load balancing (LBS), this interface is generally called when the device access service (DAS) registers the callback ENUM_DEV_SESSIONKEY_REQ

When the session is received, the interface is called (at this time, LBS notifies DAS of the SessionKey).

See also

[NET_ECMS_GetDeviceSessionKey](#)

NCMGvCn

Get device parameters.

Interface definition

```
BOOL CALLBACK NCMGvCn( IUserID, dwCommand,
    LONG
    DWORD
    NET_EHOME_CONFIG *Cn,
```

DWORD	wCn
-------	-----

);

parameter

iUserID

[IN] User ID, returned by [the registration callback function](#) registered by *NET_ECMS_StartListen* .

dwCommand

[IN] Configuration command.

Cn

{IN&OUT} Input or output configuration parameters. Different configuration commands correspond to different input or output configuration parameters.

Please refer to [the structure NET_EHOME_CONFIG](#) for the parameter format.

wCn

[IN] The size of the structure pointed to by Cn .

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is

returned, call *NET_ECMS_GetLastError* to get the error code.

Note

Different commands correspond to different conditions and output parameters.

dwCommand	ÿ pCondBuf	ÿ pOutBuf	command number
Command to get device information: <i>NET_EHOME_GET_DEVICE_INFO</i>	NULL	NET_EHOME_DEVICE_IN FO	1
Command to get version information: <i>NET_EHOME_GET_VERSION_INFO</i>	NULL	NET_EHOME_VERSION_I NFO	2
Command to get basic information of the device: <i>NET_EHOME_GET_DEVICE_CFG</i>	NULL	NET_EHOME_DEVICE_CF G	3
Command to obtain network parameters: <i>NET_EHOME_GET_NETWORK_CFG</i>	NULL	NET_EHOME_NETWORK_CFG	5

dwCommand	ÿ pCondBuf	ÿ pOutBuf	command number
Command to get encoding parameters: NET_EHOME_GET_COMPRESSION_CFG	<u>NET_EHOME_BUY_SESSION_COND</u>	<u>NET_EHOME_COMPRESS_I0N_CFG</u>	7
Command to get image parameters: NET_EHOME_GET_IMAGE_CFG		<u>NET_EHOME_IMAGE_CFG</u>	9
Command to get alarm input parameters: NET_EHOME_GET_ALARMIN_CFG	<u>NET_EHOME_ALARM_N_COND</u>	<u>NET_EHOME_ALARMIN_CFG</u>	11
Command to get the deployment plan parameters: NET_EHOME_GET_ALARM_TIME_CFG	<u>NET_EHOME_ALARM_TIME_COND</u>	<u>NET_EHOME_ALARM_TIME_ME_CFG</u>	13
Command to get alarm output parameters: NET_EHOME_GET_ALARMOUT_CFG	4-byte alarm output number, starting from <u>NET_EHOME_ALARMOUT_1</u>	<u>T_CFG</u>	15
Command to get alarm output status: NET_EHOME_GET_ALARMOUT_STATUS_CFG	4-byte alarm output number, starting from <u>NET_EHOME_ALARMOUT_1</u>	<u>T_STATUS_CFG</u>	17
Get OSD (screen menu adjustment mode) parameter command: NET_EHOME_GET_PIC_CFG	4-byte channel number	<u>NET_EHOME_PIC_CFG_22</u>	

See also

NCMvCn

NCMvCn

Set device parameters.

Interface definition

```
BOOL CALLBACK NCMvCn(
    LONG iUserID,
    DWORD dwCommand,
    NET_EHOME_CONFIG *Cn,
    DWORD );           wCn
```

parameter

iUserID

[IN] User ID, returned by the registration callback function registered by *NET_ECMS_StartListen* .

dwCommand

[IN] Configuration command.

Cn

{IN&OUT} Input or output configuration parameters. Different configuration commands correspond to different input or output configuration parameters.

Please refer to the structure *NET_EHOME_CONFIG* for the parameter format.

wCn

[IN] The size of the structure pointed to by **Cn** .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call *NET_ECMS_GetLastError* to get the error code.

Note

Different commands correspond to different conditions and output parameters.

dwCommand	ÿ pCondBuf	ÿ plnBuf command number
Command to set device information: <i>NET_EHOME_SET_DEVICE_CFG</i>	NULL	<u><i>NET_EHOME_DEVICE_CFG</i></u> <u><i>G</i></u>
Command to set network parameters: <i>NET_EHOME_SET_NETWORK_CFG</i>	NULL	<u><i>NET_EHOME_NETWORK_CFG</i></u>

dwCommand	ÿ pCondBuf	ÿ plnBuf command number
Command to set encoding parameters: NET_EHOME_SET_COMPRESSION_CFG	<u>NET_EHOME_BUY_SESSION_COND</u>	<u>NET_EHOME_COMPRESS_IOP_CFG</u> 8
Command to set image parameters: NET_EHOME_SET_IMAGE_CFG		<u>NET_EHOME_IMAGE_CFG</u> 10
Command to set alarm input parameters: NET_EHOME_SET_ALARMIN_CFG	<u>NET_EHOME_ALARM_COND</u>	<u>NET_EHOME_ALARMIN_CFG</u> 12
Command to set the deployment plan parameters: NET_EHOME_SET_ALARM_TIME_CFG	<u>NET_EHOME_ALARM_TIME_COND</u>	<u>NET_EHOME_ALARM_TIME_ME_CFG</u> 14
Command to set alarm output parameters: NET_EHOME_SET_ALARMOUT_CFG	4-byte alarm output number, starting from <u>NET_EHOME_ALARMOUT_1</u>	<u>NET_EHOME_ALARMOUT_ME_CFG</u> 16
Command to set alarm output status: NET_EHOME_SET_ALARMOUT_STATUS_CFG	4-byte alarm output number, starting from <u>NET_EHOME_ALARMOUT_1</u>	<u>NET_EHOME_ALARMOUT_STATUS_ME_CFG</u> 18
Command to set OSD (screen menu adjustment 4-byte channel number mode) parameters: NET_EHOME_GET_PIC_CFG		<u>NET_EHOME_PIC_CFG_23</u>

See also

NCMGCn

NET_ECMS_ResetDevRegisterState

Resets the registration status of devices that support ISUP version 5.0.

Interface definition

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

parameter

sDeviceID

[IN] Device ID.

return value

Returns *TRUE* for success and *FALSE* for failure.

If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

NET_ECMS_RemoteControl

Perform remote control.

Interface definition

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

parameter

lUserID

[IN] User ID, returned by *NET_ECMS_StartListen*.

dwCommand

[IN] Control command, see the table below for details.

dwCommand	describe	Command number
NET_EHOME_MANUAL_IOOUT Manually control alarm output.		19
NET_EHOME_PTZ_CTRL	Control PTZ.	1000
NET_EHOME_PRESET_CTRL	Set, delete or call preset points.	1001

dwCommand	describe	Command number
NET_EHOME_PZIN	Zoom in or out.	1002
NET_EHOME_PTRACK	3D positioning.	1003

IpCtrlParam

[IN] Control parameters, depending on the control command. For details, see [the NET_EHOME_REMOTE_CTRL_PARAM structure](#).

return value

Returns *TRUE* for success and *FALSE* for failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_SetCallback

Registers an asynchronous callback function for the Central Management Server (CMS).

Interface definition

```
BOOL CALLBACK NET_ECMS_SetCallback(
    NET_EHOME_CALLBACK_TYPE enumCallbackType,
    ECMS_CALLBACK fnCallback,
    void* );
```

parameter

enumCallbackType

[IN] Callback type. For details, see [enumeration NET_EHOME_CALLBACK_TYPE](#).

fnCallback

[IN] Callback function. For details, see [ECMS_CALLBACK](#).

nUser

[IN] User parameter pointer.

return value

Returns *TRUE* for success and *FALSE* for failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NCMxcnCBc

Exception callback function of the registry management server (CMS).

Interface definition

```
BOOL CALLBACK NCMxctinCBc(
    DWORD dwMessage,
    HANDLE hWnd, xctinCBc
    cbxcnCBc)
void *pUser
DWORD dwType,
LONG iUserID,
LONG iHandle, void *pUser

)
```

parameter

dwMessage

[IN] Message type (this parameter is only valid for Windows operating systems).

hWnd

[IN] Window handle (this parameter is valid only for Windows operating systems).

cbxcnCBc

[IN] Exception callback function. For details, see [xcnCBc](#).

nave1

[IN] User pointer.

return value

Returns *TRUE* for success and *FALSE* for failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NCMnCnC

Establish a long connection.

Interface definition

```
LONG CALLBACK NCMnCnCr(
    LONG          IUserID,
    NET_EHOME_LONG_CFG_INPUT *pLongCfgInput
)
```

parameter

IUserID

[IN] User ID.

pLongCfgInput

[IN] Long connection parameters. For details, see the [NET_EHOME_LONG_CFG_INPUT](#) structure .

Return

value A value greater than or equal to 0 indicates success, and a value less than 0 indicates failure. If the return value is less than 0, call [NET_ECMS_GetLastError](#) to obtain the error code.

NCMnCny

Destroy the persistent connection.

Interface definition

```
BOOL CALLBACK NCMnCny(
    LONG iHandle
```

parameter

iHandle

Long connection handle, [returned by NCMnCnC](#) .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NCMnCnGn

Get the session ID based on the persistent connection handle.

Interface definition

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

parameter

iHandle

Long connection handle, returned by *NCMnCnC*.

pSessionId

Session ID pointer.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call *NET_ECMS_GetLastError* to get the error code.

NCMnCnn

Send data through a persistent connection.

Interface definition

```
BOOL CALLBACK NCMnCnn(
    LONG iHandle,
    NET_EHOME_LONG_CFG_SEND *pSend )
```

parameter

iHandle

Long connection handle, returned by *NCMnCnC*.

pSend

[IN] Parameters to be sent via the long connection. For details, see the *NET_EHOME_LONG_CFG_SEND* structure.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call *NET_ECMS_GetLastError* to get the error code.

NET_ECMS_MakeIFrame

Force I-frame.

Interface definition

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

```
NET_EHOME_MAKE_I_FRAME *pIFrameParma );
```

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen* .

pIFrameParma

[IN] Parameters for forced I frame, see the structure *NET_EHOME_MAKE_I_FRAME* for details .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call *NET_ECMS_GetLastError* to get the error code.

Note:

This interface only supports ISUP5.0 protocol devices.

NET_ECMS_STDBinaryCtrl

Controls binary parameters.

Interface definition

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

parameter

IUserID

[IN] User ID.

dwCommand

[IN] Control command, see the table below for details.

Order	Command number	describe
NET_EHOME_S2C_CANCEL_SCHEDULE	0x65	Cancel the release plan.
NET_EHOME_S2C_SEND_ERRCODE 0xD		Send error code.
NET_EHOME_S2C_SET_TERMINAL_NAME	0x71	Set the terminal name.
NET_EHOME_S2C_LOGOUT	0x20	Notify the terminal to log out of the server.
NET_EHOME_S2C_RETRANSMIT_TO_TERM	0x108	Transfer data from the server to the terminal.

pSendData

[IN] Parameters used to send data. For details, see the [NET_EHOME_BINARY_SEND_DATA](#) structure.

return value

Returns *TRUE* for success and *FALSE* for failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_STDBinaryCfg

Configure binary parameters.

Interface definition

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

parameter

IUserID

[IN] User ID.

dwCommand

[IN] Configuration command, see the table below for details.

ISUPSDK (General) Development Guide

Order	Command number	describe
NET_EHOME_S2C_TIME_ADJUST	0x9	Synchronize terminal time.
NET_EHOME_S2C_SET_SERVER_INFO	0x18	Set server information.
NET_EHOME_S2C_GET_TERMINAL_INFO	0x19	Get terminal information.
NET_EHOME_S2C_POST_SCHEDULE 0x50		Issue the plan.
NET_EHOME_S2C_TERMINALCONTROL_V20	0x51	Control terminal (V20).
NET_EHOME_S2C_TERMINALCONTROL	0x53	Control terminal.
NET_EHOME_S2C_PLAYCONTROL 0x54		Control terminal playback.
NET_EHOME_S2C_SET_IPCINFO	0x55	Set the network camera information.
NET_EHOME_S2C_GET_WORKSTATUS	0x56	Get the device working status.
NET_EHOME_S2C_VERSIONUPGRADE	0x57	Upgrade your equipment.
NET_EHOME_S2C_SETPLAYERPARAM	0x58	Set the playback parameters of the terminal.
NET_EHOME_S2C_SETSERVERADDR 0x5B		Set the server information for terminal registration.
NET_EHOME_WEATHERINFO	0x61	Weather information.
NET_EHOME_S2C_GET_SCREENSHOT	0x62	Get the captured terminal screen image.
NET_EHOME_S2C_GET_DEFAULTPARAM	0x63	Get the default settings.

ISUPSDK (General) Development Guide

Order	Command number	describe
NET_EHOME_S2C_REPLACE_MATERIAL	0x64	Change the material.
NET_EHOME_S2C_SET_QUEUE_INFO	0x66	Set call management information.
NET_EHOME_S2C_PUBLISH_SHOT_PIC	0x67	Send a pop-up picture.
NET_EHOME_S2C_GET_IPCINFO	0x68	Get network camera information.
NET_EHOME_S2C_SET_TERMINAL_IP	0x69	Set the terminal IP address.
NET_EHOME_S2C_GET_TERMINAL_IP	0x70	Get the terminal IP address.
NET_EHOME_S2C_SET_SWITCHPLAN 0x100		Issue startup/shutdown plans.
NET_EHOME_S2C_GET_SWITCHPLAN	0x101	Get the power on/off schedule.
NET_EHOME_S2C_SET_VOLUME_PLAN	0x102	Set up a volume leveling schedule.
NET_EHOME_S2C_GET_VOLUME_PLAN	0x103	Get the volume leveling plan.
NET_EHOME_S2C_SET_CHARACTER 0x106		Send a text message to the terminal.
NET_EHOME_S2C_GET_CHARACTER 0x107		Get text messages from the terminal.
NET_EHOME_S2C_SET_ADB_CFG	0x110	Set ADB parameters.
NET_EHOME_S2C_GET_ADB_CFG 0x111		Get ADB parameters.
NET_EHOME_S2C_SET_TIEM_ZONE 0x112		Set the terminal time zone.
NET_EHOME_S2C_GET_TIME_ZONE 0x113		Get the terminal time zone.
NET_EHOME_S2C_SET_SADP	0x114	Set the terminal SADP information.

ISUPSDK (General) Development Guide

Order	Command number	describe
NET_EHOME_S2C_GET_SADP	0x115	Get the terminal SADP information.
NET_EHOME_S2C_GET_COMPONENT	0x116	Get terminal component information.
NET_EHOME_S2C_TRANS_WITH_RIGHT	0x10A	Transfer data between server and terminal.
NET_EHOME_S2C_PUBLISH_XML	0x200	XML data for transmission schedules, inserts, and upgrades according to.
NET_EHOME_S2C_SCREEN_SHOT 0x201		Capture the terminal screen.

pSendData

[IN] Parameters used to send data. For details, see the [NET_EHOME_BINARY_SEND_DATA](#) structure .

pRecvData

[IN] Parameters used to receive data. For details, see the [NET_EHOME_BINARY_RECV_DATA](#) structure .

return value

Returns *TRUE* for success and *FALSE* for failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_SetAliveTimeout

Set the heartbeat timeout period of ISUPSDK for a 5.0 protocol device (will not affect the heartbeat sending of the device, only ISUPSDK heartbeat detection).

Interface definition

```
BOOL CALLBACK NET_ECMS_SetAliveTimeout(
    LONG IUserID,
    DWORD dwKeepAliveSec,
    DWORD dwTimeOutCount
)
```

parameter

IUserID

[IN] User ID, returned by the [callback function registered by NET_ECMS_StartListen](#) .

dwKeepAliveSec

Single heartbeat timeout, in seconds.

dwTimeOutCount

Number of heartbeat timeouts.

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

Note:

The actual heartbeat timeout is the value of **dwKeepAliveSec** multiplied by **dwTimeOutCount**.

14.1.3 Listening Interface

NET_ECMS_StartListen

Enable the Central Management Server (CMS) listener and register a callback function to receive device registration information.

Interface definition

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

parameter

IpCMSListenPara

[IN] Listening parameters. For details, see the *NET_EHOME_CMS_LISTEN_PARAM* structure.

If the return

value is -1, it means failure, and other values indicate the handle parameter of *NET_ECMS_StopListen*. If -1 is returned, call *NET_ECMS_GetLastError* to get the error code.

NET_ECMS_StopListen

Stop the Central Management Server (CMS) from listening.

Interface definition

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

parameter

IHandle

[IN] Listener handle, returned by [NET_ECMS_StartListen](#).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_StartListenProxy

Start listening for proxy types.

Interface definition

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

parameter

IpStr

[IN] Parameters of the listening agent type. For details, see the [NET_EHOME_PT_PARAM](#) structure.

Return value

A value less than 0 indicates failure, and other values indicate the listener handle of [NET_ECMS_StopListenProxy](#). If the return value is less than 0, call [NET_ECMS_GetLastError](#) to obtain the error code.

NET_ECMS_StopListenProxy

Stop listening for a proxy type.

Interface definition

```
BOOL CALLBACK NET_ECMS_StopListenProxy(  
    LONG IListenHandle,
```

DWORD dwProxyType)

parameter

IListenHandle

[IN] Listener handle, returned by [NET_ECMS_StartListenProxy](#).

dwProxyType

[IN] Proxy type, enumerated in [NET_CMS_ENUM_PROXY_TYPE](#). The default proxy type is Hikvision private protocol proxy (ENUM_PROXY_TYPE_NETSDK).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call [NET_ECMS_GetLastError](#) to get the error code.

See also

[NET_ECMS_StartListenProxy](#)

14.1.4 Preview Interface

NET_ECMS_StartGetRealStream

Start preview.

Interface definition

```
BOOL CALLBACK NET_ECMS_StartGetRealStream(
    LONG lUserID,
    LPNET_EHOME_PREVIEWINFO_IN      pPreviewInfoln,
    LPNET_EHOME_PREVIEWINFO_OUT     PreviewInfoOut
);
```

parameter

lUserID

[IN] User ID, returned by the callback function registered by [NET_ECMS_StartListen](#).

pPreviewInfoln

[IN] Input parameters of the preview request. For details, see the [NET_EHOME_PREVIEWINFO_IN](#) structure.

pPreviewInfoOut

[OUT] Output parameters of the preview request. For details, see the [NET_EHOME_PREVIEWINFO_OUT](#) structure.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

NET_ECMS_StartGetRealStreamV11

Request to start preview.

Interface definition

```
BOOL CALLBACK NET_ECMS_StartGetRealStreamV11(
    LONG IUserID,
    NET_EHOME_PREVIEWINFO_IN_V11 *pPreviewInfoIn,
    NET_EHOME_PREVIEWINFO_OUT           *PreviewInfoOut );
```

parameter

IUserID

[IN] User ID, returned by the [callback function registered](#) by *NET_ECMS_StartListen*.

pPreviewInfoIn

[IN] Input parameters of the preview request. For details, see the [NET_EHOME_PREVIEWINFO_IN_V11 structure](#).

pPreviewInfoOut

[OUT] Output parameters of the preview request. For details, see the [NET_EHOME_PREVIEWINFO_OUT structure](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

See also

[NET_ECMS_StartPushRealStream](#)

[NET_ECMS_StopGetRealStream](#)

NET_ECMS_StartPushRealStream

The central management server (CMS) sends a request to the device, and the device starts transmitting and previewing the real-time stream.

Interface definition

```
BOOL CALLBACK NET_ECMS_StartPushRealStream(
    LONG IUserID,
```

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

parameter

iUserID

[IN] User ID, returned by [the callback function registered by NET_ECMS_StartListen](#).

pPushInfoln

[IN] Input parameters of real-time stream transmission request. For details, see the [NET_EHOME_PUSHSTREAM_IN structure](#).

pPushInfoOut

[OUT] Output parameters of the real-time code stream transmission request. For details, see the [NET_EHOME_PUSHSTREAM_OUT structure](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Note:

This interface does not need to be called for devices with ISUP protocol versions below 4.0 (not included). This interface needs to be called for devices with ISUP protocol versions above 4.0 (inclusive). Before calling this interface, ensure that [NET_ECMS_StartGetRealStreamV11](#) or [NET_ECMS_StartGetRealStream](#) has been called successfully and that CMS has received the device session ID in the preview request response.

See also

[NET_ECMS_StartGetRealStreamV11](#)

NET_ECMS_StopGetRealStream

Request to stop preview.

Interface definition

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

parameter

iUserID

[IN] User ID, returned by [the callback function registered by NET_ECMS_StartListen](#).

iSessionID

[IN] Session ID, returned by [NET_ECMS_StartGetRealStreamV11](#) or [NET_ECMS_StartGetRealStream](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_StopGetRealStreamEx

Request to stop preview. This interface supports asynchronous mode.

Interface definition

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

parameter

IUserID

[IN] User ID, returned by the callback function registered by [NET_ECMS_StartListen](#).

pStopParam

[IN] Request parameters for stopping preview. For details, see the [NET_EHOME_STOPSTREAM_PARAM](#) structure.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Extended from

[NET_ECMS_StopGetRealStream](#)

14.1.5 Playback Interface

NET_ECMS_StartFindFile

Start searching for files.

Interface definition

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

parameter

IUserID

[IN] User ID, returned by [the callback function registered by *NET_ECMS_StartListen*](#).

pFindCond

[IN] Search parameters, see [NET_EHOME_FINDCOND definition](#).

Return value

-1 indicates failure, and other values indicate the handles of [NET_ECMS_FindNextFile_V11](#) and [NET_ECMS_StopFindFile](#). If -1 is returned, call

[NET_ECMS_GetLastError](#) to get [the error code](#).

Note:

This interface has been replaced by [NET_ECMS_StartFindFile_V11](#). It is recommended to use a higher version interface.

NET_ECMS_StartFindFile_V11

Start searching for files.

Interface definition

```
LONG CALLBACK NET_ECMS_StartFindFile_V11(
    LONG IUserID,
    LONG ISearchType,
    VOID *pFindCond,
    DWORD dwCondSize );
```

parameter

IUserID

[IN] User ID, returned by [the registration callback function registered by *NET_ECMS_StartListen*](#).

ISearchType

[IN] Search type, see the enumeration definition [SEARCH_TYPE](#).

pFindCond

[IN] Buffer for saving search conditions. Different search types correspond to different search conditions. For details, see the notes.

dwCondSize

[IN] Retrieve condition buffer size.

return value

Returning -1 indicates failure, and returning other values indicates success of NET_ECMS_FindNextFile_V11 and NET_ECMS_StopFindFile. Handle.

If -1 is returned, call NET_ECMS_GetLastError to get the error code.

Remark

The relationship between the search type (ISearchBar) and the search condition (pFindCond) is shown in the following table.

ISearchBar	pFindCond
ENUM_SEARCH_RECORD_FILE	<u>NET_EHOME_REC_FILE_COND</u>
ENUM_SEARCH_PICTURE_FILE	<u>NET_EHOME_PIC_FILE_COND</u>
ENUM_SEARCH_FLOW_INFO	<u>NET_EHOME_FLOW_COND</u>
ENUM_SEARCH_DEV_LOG	<u>NET_EHOME_DEV_LOG_COND</u>
ENUM_SEARCH_ALARM_HOST_LOG	<u>NET_EHOME_ALARM_HOST_LOG_COND</u>

NET_ECMS_FindNextFile

Get the retrieved files one by one (only supports video file query).

Interface definition

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

parameter

IHandle

[IN] Retrieval handle returned by NET_ECMS_StartFindFile.

pFindData

[OUT] Returned data structure, see NET_EHOME_FINDDATA definition.

Return

value -1 indicates failure, and other values indicate acquisition status. For details, see the following table. *If -1 is returned, call NET_ECMS_GetLastError to obtain the error code.*

Status Name	status code	describe
ENUM_GET_NEXT_STATUS_SUCCESS	1000	A file is obtained. After processing the data, call the interface again to obtain the next file.
ENUM_GET_NETX_STATUS_NO_FILE	1001	File not found.
ENUM_GET_NETX_STATUS_NEED_WAIT	1002	Retrieving. Please wait.
ENUM_GET_NEXT_STATUS_FINISH	1003	No more files. Ending search.
ENUM_GET_NEXT_STATUS_FAILED	1004	Retrieval exception.
ENUM_GET_NEXT_STATUS_NOT_SUPPORT	1005	The device does not support the operation.

Note:

This interface has been replaced by NET_ECMS_FindNextFile_V11. It is recommended to use a higher version interface.

NET_ECMS_FindNextFile_V11

Get the retrieved files one by one.

Interface definition

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

parameter

IHandle

[IN] Retrieval handle returned by NET_ECMS_StartFindFile_V11.

pFindData

[OUT] Returned data structure. Different search types (SearchType) configured in NET_ECMS_StartFindFile_V11 correspond to different data structures. For details, see the table below.

SearchType	pFindData
Retrieve video file: ENUM_SEARCH_RECORD_FILE	<u>NET_EHOME_REC_FILE</u>
Retrieve picture files: ENUM_SEARCH_PICTURE_FILE	<u>NET_EHOME_PIC_FILE</u>
Retrieve flow information: ENUM_SEARCH_FLOW_INFO	<u>NET_EHOME_FLOW_INFO</u>
Retrieve device log: ENUM_SEARCH_DEV_LOG	<u>NET_EHOME_DEV_LOG</u>
Retrieve alarm host log: ENUM_SEARCH_ALARM_HOST_LOG	<u>NET_EHOME_ALARM_HOST_LOG</u>

dwDataSize

[OUT] The size of the returned data.

return value

- *Returning -1 indicates failure, and returning other values indicates the acquisition status. See the table below for details.*

Status Name	status code	describe
ENUM_GET_NEXT_STATUS_SUCCESS	1000	Get a file. After processing the file, call the interface again to obtain the next file.
ENUM_GET_NETX_STATUS_NO_FILE	1001	File not found.
ENUM_GET_NETX_STATUS_NEED_WAIT	1002	Retrieving. Please wait.
ENUM_GET_NEXT_STATUS_FINISH	1003	No more files. Ending search.
ENUM_GET_NEXT_STATUS_FAILED	1004	Retrieve exceptions.
ENUM_GET_NEXT_STATUS_NOT_SUPPORT	1005	The device does not support the operation.

- *If -1 is returned, please call NET_ECMS_GetLastError to get the error code.*

Note :

- This interface can only be called after calling [NET_ECMS_StartFindFile_V11](#) to set the search conditions and obtain the search handle.
- To obtain all search results, this interface must be called repeatedly.

NET_ECMS_StopFindFile

Stops retrieving files and releases resources.

Interface definition

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

parameter

iHandle

[IN] Retrieval handle returned by [NET_ECMS_StartFindFile_V11](#) .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_StartPlayBack

Request to start playback.

Interface definition

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

parameter

iUserID

[IN] User ID, returned by the callback function registered by [NET_ECMS_StartListen](#) .

pPlayBackInfoln

[IN] Input parameters of the playback request. For details, see the [NET_EHOME_PLAYBACK_INFO_IN](#) structure .

pPlayBackInfoOut

[OUT] Output parameters of the playback request. For details, see the [NET_EHOME_PLAYBACK_INFO_OUT structure](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

See also

[NET_ECMS_StartPushPlayBack](#)

[NET_ECMS_StopPlayBack](#)

NET_ECMS_StartPushPlayBack

The central management server (CMS) sends a request to the device, and the device starts transmitting the playback stream.

Interface definition

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

parameter

IUserID

[IN] User ID, returned by [the callback function registered](#) by [NET_ECMS_StartListen](#).

pPushInfoIn

[IN] Input parameters of playback stream transmission request. For details, see the [NET_EHOME_PUSHPLAYBACK_IN structure](#).

pPushInfoOut

[OUT] Output parameters of playback stream transmission request. For details, see the [NET_EHOME_PUSHPLAYBACK_OUT structure](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Note

For devices that support ISUP version 4.0 and above, before calling this interface, make sure that the CMS has received the device session ID in the response to the playback request.

See also

[NET_ECMS_StartPlayBack](#)

NET_ECMS_PlayBackOperate

Control playback, such as pausing and resuming playback.

Interface definition

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

parameter

iUserID

[IN] User ID, returned by [the callback function registered by NET_ECMS_StartListen](#) .

enumMode

[IN] Control operation type, enumerated in [NET_EHOME_PLAYBACK_OPERATE_MODE](#) .

pOperateParam

[IN] Control operation parameters, depending on the control operation type (enumMode). For details on the parameters for pausing or resuming playback, see the [NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM](#) structure .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_StopPlayBack

Request to stop playback.

Interface definition

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

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen*.

iSessionID

[IN] Session ID, returned by *NET_ECMS_StartPlayBack*.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

*is returned, call *NET_ECMS_GetLastError* to get the error code.*

NET_ECMS_StopPlayBackEx

Request to stop playback. This interface supports asynchronous mode.

Interface definition

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

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen*.

pStopParam

[IN] Request parameters for stopping playback. For details, see the *NET_EHOME_STOPPLAYBACK_PARAM* structure.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

*is returned, call *NET_ECMS_GetLastError* to get the error code.*

Extended from

NET_ECMS_StopPlayBack

14.1.6 Voice Intercom Interface**NET_ECMS_StartVoiceWithStmServer**

Request to start voice intercom via streaming server (SMS).

Interface definition

```
BOOL CALLBACK NET_ECMS_StartVoiceWithStmServer( IUserID,
    LONG
    NET_EHOME_VOICE_TALK_IN *IpVoiceTalkIn
    NET_EHOME_VOICE_TALK_OUT *IpVoiceTalkOut );
```

parameter

IUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen*.

IpVoiceTalkIn

[IN] Input parameters of the voice intercom request. For details, see the *NET_EHOME_VOICE_TALK_IN* structure.

IpVoiceTalkOut

[OUT] Output parameters of the voice intercom request. For details, see the *NET_EHOME_VOICE_TALK_OUT* structure.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

*is returned, call *NET_ECMS_GetLastError* to get the error code.*

See also

NET_ECMS_StopVoiceTalkWithStmServer**NET_ECMS_StartPushVoiceStream**

The Central Management Server (CMS) sends a request to the device, and the device starts transmitting the audio code stream for voice intercom.

Interface definition

```
BOOL CALLBACK NET_ECMS_StartPushVoiceStream(
    LONG IUserID,
    NET_EHOME_PUSHVOICE_IN *IpPushParamIn,
    NET_EHOME_PUSHVOICE_OUT *IpPushParamOut );
```

parameter

IUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen* .

IpPushParamIn

[IN] Input parameters of the audio stream transmission request. For details, see the *NET_EHOME_PUSHVOICE_IN* structure .

IpPushParamOut

[OUT] Output parameters of the audio stream transmission request. For details, see the structure *NET_EHOME_PUSHVOICE_OUT* .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call *NET_ECMS_GetLastError* to get the error code.

Note

For devices that support ISUP version 4.0 and above, before calling this interface, make sure that CMS has received the device session ID in the voice intercom request response.

See also

NET_ECMS_StartVoiceWithStmServer

NET_ECMS_StopVoiceTalkWithStmServer

Request to stop voice intercom via streaming server (SMS).

Interface definition

```
BOOL CALLBACK NET_ECMS_StopVoiceTalkWithStmServer(
    LONG IUserID,
    LONG ISessionID );
```

parameter

IUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen* .

ISessionID

[IN] Session ID, returned by *NET_ECMS_StartVoiceWithStmServer* .

Return

value *TRUE* indicates success, *FALSE* indicates failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_StopVoiceTalkWithStmServerEx

Request to stop voice intercom via streaming media server (SMS). This interface supports asynchronous mode.

Interface definition

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

parameter

iUserID

[IN] User ID

pStopParam

[IN] Request parameters for stopping voice intercom via SMS. For details, see the structure

[NET_EHOME_STOPVOICETALK_STM_PARAM](#)

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Extended from

[NET_ECMS_StopVoiceTalkWithStmServer](#)

NET_ECMS_StartVoiceTalk

Start voice intercom or forward audio.

Interface definition

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

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen*.

dwVoiceChan

[IN] Voice intercom channel number.

pVoiceTalkPara

[IN] Voice talk or audio forwarding parameters. For details, see the *NET_EHOME_VOICETALK_PARA* structure.

Return

value -1 indicates failure, and other values indicate the handle of *NET_ECMS_StopVoiceTalk*. If -1 is returned, call *NET_ECMS_GetLastError* to get the error code.

Note:

For Windows 7 operating system, if there is no external audio device, this interface will return -1.

See also

NET_ECMS_SendVoiceTransData

NET_ECMS_SendVoiceTransData

Forwards audio data to the device.

Interface definition

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

parameter

VoiceHandle

[IN] Audio forwarding handle, returned by *NET_ECMS_StartVoiceTalk*.

pSendBuf

[IN] Pointer to the buffer that stores the audio data.

dwBufSize

[IN] The size of the data in the buffer.

Return

value *TRUE* indicates success, *FALSE* indicates failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ECMS_StopVoiceTalk

Stop voice intercom or forward audio.

Interface definition

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

parameter

VoiceHandle

[IN] Handle of voice intercom or audio forwarding, returned by [NET_ECMS_StartVoiceTalk](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call [NET_ECMS_GetLastError](#) to get the error code.

14.1.7 Message Transmission Interface

NCMXMCn

Transmit ISUP commands and messages.

Interface definition

```
BOOL CALLBACK NCMXMCn(
    LONG lUserID,
    NET_EHOME_XML_CFG *pXmlCfg,
    DWORD wCn );
```

parameter

lUserID

[IN] User ID, returned by the callback function registered by [NET_ECMS_StartListen](#).

pXmlCfg

[IN] The configuration parameters to be transmitted. Different configuration commands correspond to different configuration parameters. For details, see the structure.

NET_EHOME_XML_CFG

wCn

[IN] The size of the structure pointed to by **pXmlCfg**.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, please call [NET_ECMS_GetLastError](#) to get the error code. Note:

During

the ISUP command and message transmission process, data is only sent and received, and no verification is performed on the data.

NCMXMCnx

Set the ISAPI (Intelligent Security Access API) request URL and message transmission parameters. This interface supports asynchronous callback mode.

Interface definition

```
BOOL CALLBACK NCMXMCnx(
    LONG IUserID,
    NET_EHOME_XML_CFG *pXmlCfg,
    DWORD *pHandle
)
```

parameter

IUserID

[IN] User ID.

pXmlCfg

[IN] ISAPI request URL and message transmission parameters. For details, see the [NET_EHOME_XML_CFG](#) structure.

Outside

[OUT] Asynchronous callback handle, corresponding to dwHandle in the callback function

[ASYNC_RESPONSE_CB](#) registered by *NCMmCnRnCB*.

Return

value *TRUE* indicates success, *FALSE* indicates failure.

If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Extended from

NCMMCn

NET_ECMS_XMLRemoteControl

Transmits ISUP (Intelligent Security Uplink Protocol) control commands and XML messages.

Interface definition

```
BOOL CALLBACK NET_ECMS_XMLRemoteControl(
    LONG iUserID,
    NET_EHOME_XML_REMOTE_CTRL_PARAM *IpCtrlParam,
    DWORD dwCtrlSize
```

parameter

iUserID

[IN] User ID, returned by the [registration callback function \(DEVICE_REGISTER_CB \)](#) registered by [NET_ECMS_StartListen](#).

IpCtrlParam

[IN] Control parameters. For details, see [NET_EHOME_XML_REMOTE_CTRL_PARAM](#).

dwCtrlSize

[IN] The size of the buffer that stores the control parameters. The size of the buffer is equal to the size of the [NET_EHOME_XML_REMOTE_CTRL_PARAM](#) structure.

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the [error code](#).

Note: This

interface does not support the transmission of get or query commands.

NCMXMCn

Transmit the request URL and the DELETE method.

Interface definition

```
BOOL CALLBACK NCMXMCn(
    LONG iUserID,
    NET_EHOME_PTXML_PARAM *IpPTXMLParam );
```

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen* .

IpPTXMLParam

[IN] Parameters to be transmitted. For details, see the *NET_EHOME_PTXML_PARAM* structure .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

*is returned, call *NET_ECMS_GetLastError* to get the error code.*

Note:

This interface only receives or sends data and does not verify or modify the data, so you must ensure the correctness of the data yourself.

NCMGXMCn

Transmits the request URL and GET method.

Interface definition

```
BOOL CALLBACK NCMGXMCn(  
    LONG iUserID,  
    NET_EHOME_PTXML_PARAM *IpPTXMLParam );
```

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen* .

IpPTXMLParam

[IN] Parameters to be transmitted. For details, see the *NET_EHOME_PTXML_PARAM* structure .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

*is returned, call *NET_ECMS_GetLastError* to get the error code.*

Note:

This interface only receives or sends data and does not verify or modify the data, so you must ensure the correctness of the data yourself.

NCMXMCn

Transmits the request URL and POST method.

Interface definition

```
BOOL CALLBACK NCMXMCn(
    LONG iUserID,
    NET_EHOME_PTXML_PARAM *IpPTXMLParam );
```

parameter

iUserID

[IN] User ID, returned by [the callback function registered by NET_ECMS_StartListen](#) .

IpPTXMLParam

[IN] Parameters to be transmitted. For details, see the [NET_EHOME_PTXML_PARAM structure](#) .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call NET_ECMS_GetLastError to get the error code.

Note:

This interface only receives or sends data and does not verify or modify the data, so you must ensure the correctness of the data yourself.

NCMXMCn

Transfer request URL and PUT method.

Interface definition

```
BOOL CALLBACK NCMXMCn(
    LONG iUserID,
    NET_EHOME_PTXML_PARAM *IpPTXMLParam );
```

parameter

iUserID

[IN] User ID, returned by [the callback function registered by NET_ECMS_StartListen](#) .

IpPTXMLParam

[IN] Parameters to be transmitted. For details, see the [NET_EHOME_PTXML_PARAM structure](#) .

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

Note:

This interface only receives or sends data and does not verify or modify the data, so you must ensure the correctness of the data yourself.

NCMCn

Stream HTTP pages.

Interface definition

```
BOOL CALLBACK NCMCn( iUserID,
    LONG
    LPNET_EHOME_HTTP_PARAM IpParam )
```

parameter

iUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen*.

IpParam

HTTP page transmission parameters, see *NET_EHOME_HTTP_PARAM*.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

Note:

Nodes in the XML_DeviceCap message returned by GET ymcbt
rrDetermine whether the device supports ISUPV5.0HTTP page transparent transmission.

NET_ECMS_ISAPIPassThrough

The transmission command includes the request URL and operation method (such as GET, PUT, POST, DELETE).

Interface definition

```
BOOL CALLBACK NET_ECMS_ISAPIPassThrough(
    LONG IUserID,
```

```
NET_EHOME_PTXML_PARAM *IpParam );
```

parameter

IUserID

[IN] User ID, returned by the callback function registered by *NET_ECMS_StartListen*.

IpParam

[IN] Parameters to be transmitted. For details, see the *NET_EHOME_PTXML_PARAM* structure.

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

NET_ECMS_SendPassthroughData

Use HTTP or a proprietary protocol to transfer data.

Interface definition

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

parameter

B

[IN] Buffer to store the data to be transferred.

dwDataLen

[IN] The size of the data to be transferred.

dwProxyType [IN]

Proxy type. The default value is "ENUM_PROXY_TYPE_NETSDK" (private protocol proxy).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned,

call *NET_ECMS_GetLastError* to get the error code.

NET_ECMS_SetPassthroughDataCallback

Registers a callback function to receive transmission data (via HTTP or private protocol).

Interface definition

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

parameter

fnPassthroughDataCb

[IN] Callback function pointer. For details, see [PASSTHROUGHDATA CALLBACK](#).

nUser

[IN] User pointer.

dwProxyType

[IN] Proxy type. The default value is ENUM_PROXY_TYPE_NETSDK (private protocol proxy).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

NCMXmCnRnCB

Registers a response callback function for transmitting ISAPI messages.

Interface definition

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

parameter

fnCB

[IN] Response callback function. For details, see [ASYNC_RESPONSE_CB](#).

nUser

[IN] User parameter pointer.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call *NET_ECMS_GetLastError* to get the error code.

NCMCnvc

Converts types between HTTP protocol and private protocol.

Interface definition

```
LONG CALLBACK NCMCnvrcc( void *cB, dwSrcBufLen, void *B,
dwDestBufLen,
DWORD
DWORD
NET_EHOME_PASSTHROUGH_PARAM *IpParam,
BOOL          bToPassthrough
)
```

parameter

cB

[IN] Input buffer.

dwSrcBufLen

[IN] Input buffer size.

B

[IN] Output buffer.

dwDestBufLen

[IN] Size of the output buffer.

IpParam

[IN] Protocol parameters. For details, see the *NET_EHOME_PASSTHROUGH_PARAM* structure .

bToPassthrough

[IN] Protocol conversion direction. If the value of this parameter is "TRUE", the HTTP protocol starts to be converted to the private protocol for transmission;

otherwise, the private protocol starts to be converted to the HTTP protocol for transmission.

The return

value is -1 if it fails, and other values indicate the size of the converted data.

If -1 is returned, call NET_ECMS_GetLastError to get the error code.

14.2 Alarm Monitoring Module

14.2.1 Basic Interface

NET_EALARM_Init

Initializes the alarm repository of the Alarm Management Server (AMS).

Interface definition

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

Return value

TRUE indicates success, *FALSE* indicates failure.

Remarks

- If initialization fails, please check the path of the loaded library file and the system environment. •

This interface must be called in conjunction with NET_EALARM_Fini.

NET_EALARM_Fini

Deinitializes the alarm library and releases the resources occupied by the Alarm Management Server (AMS).

Interface definition

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

Return value

TRUE indicates success, *FALSE* indicates failure.

Note:

This interface must be called together with NET_EALARM_Init.

NET_EALARM_SetSDKInitCfg

Set the initialization parameters of the Alarm Management Server (AMS).

Interface definition

```
BOOL CALLBACK NET_EALARM_SetSDKInitCfg(
    NET_EHOME_EALARM_INIT_CFG_TYPE enumType, void* const nB
)
```

parameter

enumType

[IN] Initialization configuration type, see enumeration [NET_EHOME_EALARM_INIT_CFG_TYPE](#) for details .

nB

[IN] Initialize configuration parameters, depending on the configuration type (enumType), see the corresponding relationship below.

enumType	nB
NET_EHOME_EALARM_INIT_CFG_LIBEAY_ PATH	Path with the name of the OpenSSL cryptographic library.
NET_EHOME_EALARM_INIT_CFG_SSLEAY_ PATH	Path with the library name of the OpenSSL communication library.

Return value

TRUE indicates success, *FALSE* indicates failure.

Remark

- This interface must be called before initialization, otherwise the settings will not take effect. • If the call fails, the error code cannot be obtained or the correct error code cannot be obtained, because the error code must be called after initialization. Obtain.
- This interface does not check the validity of the OpenSSL path. The path is verified when the library is loaded. If the path is incorrect, other The relevant interfaces will not report errors, but errors will be written to the log.
- When multiple modules are used in the same process, it is recommended to set the same OpenSSL path for all modules. For example, if the registration module (CMS) and the alarm management module (AMS) are used in the same process, if CMS has loaded OpenSSL, but AMS

If the OpenSSL path of AMS is different from that of CMS and the version number of the library is different, AMS will use the loaded OpenSSL.

- Different modules (AMS/CMS/SS/SMS) are located in the same process. When loading the OpenSSL library, ensure that different modules use The OpenSSL library path used is valid, and the OpenSSL library version and library name are the same.

NET_EALARM_GetBuildVersion

Get the version information of the Alarm Management Server (AMS).

Interface definition

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

The return

value returns the version number. The upper two bytes represent the major version, and the lower two bytes represent the minor version, for example: 0x02040001 (version 2.4.0.1).

NET_EALARM_SetLogFile

Set parameters to enable the logging function of the Alarm Management Server (AMS).

Interface definition

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

parameter

nLogLevel

[IN] Log type. 0-Enable logging (default), 1-Error log, 2-Error and debug log, 3-Error, debug and information log.

strLogDir

[IN] Log file save path. The default save path in Windows operating system is "C:\\\$dkLog\\\"", and the default save path in Linux operating system is "/home/sdklog/". If you want to use the default save path, set this parameter to "NULL".

bAutoDel

[IN] Whether to automatically delete the log file after a certain period of time: "TRUE" - yes (default), "FALSE" - no.

When "FALSE", log compression will be enabled. Every time 50 log files are generated, log compression will be performed (the 50 generated log files will be compressed into one compressed file).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call

[NET_EALARM_GetLastError](#) to get the error code.

Notes •

The path to save the log file must be an absolute path and must end with "\\", for example: "C:\\SdkLog\\\". • If the save path is changed, the changed path will be used to save the next log file. • When the value of **bAutoDel** is "TRUE", the first log file will not be deleted because the file contains the start time.

- Allow HCISUPSDK to modify log parameters during runtime. You can add the HCISUPSDK_Log_Switch.xml file in the runtime directory to configure the runtime log parameters.

[Log parameter configuration](#)

NET_EALARM_GetLastError

If the call fails or completes, you can get the error code.

Interface definition

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

Return value

The return value is an error code. For details, see [ISUPSDK error code](#)

NET_EALARM_GetSDKLocalCfg

Gets the local configuration parameters of the Alarm Management Server (AMS).

Interface definition

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

parameter

enumType

[IN] Supported local configuration type macro definition values: 0, 4 and 5, see enumeration definition [NET_EHOME_LOCAL_CFG_TYPE](#)

B

[OUT] Local configuration parameters, related to the local configuration type (enumType), see the table below for details.

enumType	B	illustrate
UNDEFINE	Undefined	Undefined.
ACTIVE_ACCESS_SECURITY	NET_EHOME_LOCAL_ACCESS_SECURITY Get AMS module device active	Access security information.
LOCAL_CFG_TYPE_GENERAL	NET_EHOME_LOCAL_GENERAL_CFG Get the local general CFG of AMS module	Configuration.
COM_PATH	COM path, pointing to a string buffer (for example: "./HCAapSDKCom/").	Get the HCAapSDKCom of the AMS module (for HCISUP related dependency library directory path).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned,

call [NET_EALARM_GetLastError](#) to get the error code.

See also

[NET_EALARM_SetSDKLocalCfg](#)

NET_EALARM_SetSDKLocalCfg

Set local configuration parameters for the Alarm Management Server (AMS).

Interface definition

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

parameter

enumType

[IN] Supported local configuration type macro definition values: -1, 0, 1, 2, 3 and 4, see enumeration definition

[NET_EHOME_LOCAL_CFG_TYPE](#)

nB

[IN] Local configuration parameters, related to the local configuration type (enumType), see the table below for details.

enumType	B	illustrate
UNDEFINE	Undefined	Undefined.
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u> sets the AMS module device to take the initiative Access security information.	
LOCAL_CFG_TYPE_GENERAL	<u>NET_EHOME_LOCAL_GENERAL_CFG</u> sets the local general purpose of the AMS module Configuration.	
COM_PATH	COM path, pointing to a string buffer (for example: "./HCAapSDKCom").	Set the HCAapSDKCom of the AMS module (for HCISUP related dependency library directory path).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call

[NET_EALARM_GetLastError](#) to get the error code.

See also

[NET_EALARM_GetSDKLocalCfg](#)

14.2.2 Configuration and Control Interface

[NET_EALARM_GetDeviceSessionKey](#)

Get the session key information of the device.

Interface definition

```
BOOL CALLBACK NET_EALARM_GetDeviceSessionKey(
    NET_EHOME_DEV_SESSIONKEY *pDeviceKey )
```

parameter

pDeviceKey

[OUT] Session key information. See [NET_EHOME_DEV_SESSIONKEY](#) for details .

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_EALARM_GetLastError](#) to get the error code.

See also

[NET_EALARM_SetDeviceSessionKey](#)

NET_EALARM_SetDeviceSessionKey

Set the session key for the device.

Interface definition

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

parameter

deviceKey

[IN] Session key information for devices supporting ISUP version 5.0. For details, see [NET_EHOME_DEV_SESSIONKEY](#).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_EALARM_GetLastError](#) to get the error code.

Remarks

- Used to set SessionKey when the AMS module authenticates the ISUP5.0 protocol device. •

The device's SessionKey is generated when the CMS module registers and calls back [ENUM_DEV_SESSIONKEY](#). The AMS module needs to be notified of the SessionKey used by the device. The AMS module sets the SessionKey used for device authentication by calling this interface. Only when the correct SessionKey is set can the ISUP5.0 protocol device access the AMS normally and push alarm information.

See also

[NET_EALARM_GetDeviceSessionKey](#)

14.2.3 Listening Interface

NET_EALARM_StartListen

Enable Alarm Management Server (AMS) listening and register callback functions to receive alarm information.

Interface definition

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

parameter

pAlarmListenParam

[IN] Input parameter used to receive alarm information after monitoring is enabled. For details, see the structure

[NET_EHOME_ALARM_LISTEN_PARAM](#)

Return

value *-1* indicates failure, and other values indicate the [handle of NET_EALARM_StopListen](#). If *-1* is returned, call [NET_EALARM_GetLastError](#) to get the error code.

NET_EALARM_StopListen

Stops monitoring of the Alarm Management Server (AMS).

Interface definition

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

parameter

lListenHandle

[IN] Alarm listener handle, returned by [NET_EALARM_StartListen](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_EALARM_GetLastError](#) to get the error code.

14.3 Flow acquisition module

14.3.1 Basic Interface

NET_ESTREAM_Init

Initialize the stream module library.

Interface definition

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

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call

NET_ECMS_GetLastError to get the error code.

Note:

This interface must be called together with *NET_ESTREAM_Fini*.

NET_ESTREAM_End

Release the resources occupied by the stream fetching module.

Interface definition

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

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call

NET_ECMS_GetLastError to get the error code.

Note:

This interface must be called together with *NET_ESTREAM_Init*.

NET_ESTREAM_SetSDKInitCfg

Set the initialization parameters of the streaming media server (SMS).

Interface definition

```
BOOL CALLBACK NET_ESTREAM_SetSDKInitCfg(
    NET_EHOME_ESTREAM_INIT_CFG_TYPE enumType, void* const nB
)
```

parameter

enumType

[IN] Initialization configuration type, see enumeration [NET_EHOME_ESTREAM_INIT_CFG_TYPE](#) for details .

nB

[IN] Initialization parameters, depends on the configuration type, see details below.

enumType	nB
NET_EHOME_ESTREAM_INIT_CFG_LIBEAY_ PATH	Path with the name of the OpenSSL cryptographic library.
NET_EHOME_ESTREAM_INIT_CFG_SSLEAY_ PATH	Path with the library name of the OpenSSL communication library.

Return value

TRUE indicates success, *FALSE* indicates failure.

Note : •

This interface must be called before initialization, otherwise the settings will not take effect. • If the call fails, the error code cannot be obtained or the correct error code cannot be obtained, because the error code must be initialized before it can be used. Obtain.

- This interface does not check the validity of the OpenSSL path. The path is verified when the library is loaded. If the path is incorrect, other relevant interfaces will not report errors, but errors will be written to the log.
- When multiple modules are used in the same process, it is recommended to set the same OpenSSL path for all modules. For example, if the Registration Module (CMS) and the Alarm Management Module (AMS) are used in the same process, if the CMS has OpenSSL loaded, but the OpenSSL path of AMS is different from that of CMS, and the version number of the library is also different, AMS will use the loaded OpenSSL.
- Different modules (AMS/CMS/SS/SMS) are located in the same process. When loading the OpenSSL library, ensure that different modules use the OpenSSL library path used is valid, and the OpenSSL library version and library name are the same.

NET_ESTREAM_GetBuildVersion

Get the version information of the streaming media server (SMS).

Interface definition

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

The return

value returns the version number. The upper two bytes represent the major version, and the lower two bytes represent the minor version, for example: 0x02040001 (version 2.4.0.1).

NET_STREAM_SetLogFile

Set the parameters to enable the streaming media server (SMS) log function.

Interface definition

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

parameter

nLogLevel

[IN] Log type: 0-Enable logging (default), 1-Error log, 2-Error and debug log, 3-Error, debug and information log.

strLogDir

[IN] Log file save path. The default save path in Windows operating system is "C:\\\\SdkLog\\\\", and the default save path in Linux operating system is "/home/sdklog/". If you want to use the default save path, set this parameter to "NULL".

bAutoDel

[IN] Whether to automatically delete the log file after a certain period of time: "TRUE" - yes (default), "FALSE" - no.

When "FALSE", log compression will be enabled. Every time 50 log files are generated, log compression will be performed (the 50 generated log files will be compressed into one compressed file).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call

`NET_ESTREAM_GetLastError` to get the error code.

Notes •

The path to save the log file must be an absolute path and must end with "\\", for example: "C:\\SdkLog\\". • If the save path is changed, the changed path will be used to save the next log file. • When the value of **bAutoDel** is "TRUE", the first log file will not be deleted because the file contains the start time.

- Allow HCISUPSDK to modify log parameters during runtime. You can add the HCISUPSDK_Log_Switch.xml file in the runtime directory to configure the runtime log parameters.

[Log parameter configuration](#)

NET_ESTREAM_GetLastError

If the stream acquisition fails or is completed, get the error code.

Interface definition

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

Return value

The return value is an error code. For details, see [ISUPSDK error code](#)

NET_ESTREAM_GetSDKLocalCfg

Get the local configuration parameters of the streaming media server (SMS).

Interface definition

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

parameter

enumType

[IN] Supported local configuration type macro definition values: 0 and 5, see enumeration definition [NET_EHOME_LOCAL_CFG_TYPE](#).

B

[OUT] Local configuration parameters, related to the local configuration type (enumType), see the table below for details.

enumType	B
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u>
COM_PATH	COM path.

return value

Returns *TRUE* for success and *FALSE* for failure.

If *FALSE* is returned, call NET_ESTREAM_GetLastError to get the error code.

See also

NET_ESTREAM_SetSDKLocalCfg

NET_ESTREAM_SetSDKLocalCfg

Set local configuration parameters of the streaming media server (SMS).

Interface definition

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

parameter

enumType

[IN] Supported local configuration type macro definition value: 0, see enumeration definition NET_EHOME_LOCAL_CFG_TYPE.

nB

[IN] Local configuration parameters, related to the local configuration type (enumType), see the table below for details.

enumType	B	illustrate
UNDEFINE	Undefined.	Undefined.
ACTIVE_ACCESS_SECURITY	<u>NET_EHOME_LOCAL_ACCESS_SECURITY</u> sets the device home page of the SMS module.	Automatic access security.

enumType	B	illustrate
COM_PATH	COM path, pointing to a string buffer. Gets the SMS module	HCAapSDKCom (for Path to the directory of HCISUP related dependency libraries).
STREAM_PLAYBACK_PARAM	<u>NET_EHOME_LOCAL_PLAYBACK_PARA</u> <u>M</u>	Set whether to enable playback synchronous reception.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ESTREAM_GetLastError* to get the error code.

Remark

[NET_ESTREAM_GetSDKLocalCfg](#)

NRMxcnCBc

Register a callback function to receive exception information.

Interface definition

```
BOOL CALLBACK NRMxctinCBc( dwMessage, hWnd, xctinCBc
    DWORD      cbxcnCBc,
    HANDLE     void
    *pUser );
```

parameter

dwMessage

[IN] Reserved.

hWnd

[IN] The window handle used to receive exception information.

cbxcnCBc

[IN] Callback function that receives exception information. For [details, see xcnCBc](#).

navel

[IN] User parameters.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE*

is returned, call *NET_ECMS_GetLastError* to get the error code.

14.3.2 Preview Interface

NET_ESTREAM_StartListenPreview

Start the streaming media server (SMS) listening and register a callback function to receive device connection requests.

Interface definition

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

parameter

pListenParam

[IN] Preview listener parameters. For details, see the *NET_EHOME_LISTEN_PREVIEW_CFG* structure.

Return

value -1 indicates failure, and other values indicate the handle of *NET_ESTREAM_StopListenPreview*. If -1 is

returned, call *NET_ESTREAM_StopListenPreview* to get the error code.

NET_ESTREAM_StopListenPreview

Stops the preview monitoring of the streaming media server (SMS) and disconnects from the device.

Interface definition

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

parameter

IListenHandle

[IN] Preview listener handle, returned by *NET_ESTREAM_StartListenPreview*.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Note:

After calling this interface, all links of the listening port will be disconnected.

NET_ESTREAM_StopPreview

Stop forwarding a certain channel to preview the real-time stream.

Interface definition

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

parameter

IPreviewHandle

[IN] Preview listener handle, returned by the callback function [PREVIEW_NEWSLINK_CB](#) registered by [NET_ESTREAM_StartListenPreview](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Note:

For a listening port associated with multiple channels, after calling this interface, the preview of only one of the channels will be stopped.

NET_ESTREAM_SetPreviewDataCB

Register a callback function to receive the real-time stream.

Interface definition

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

parameter

iHandle

[IN] Preview handle, returned by the callback function [PREVIEW_NEWINLINK_CB](#) registered by [NET_ESTREAM_StartListenPreview](#).

pStruCBParam

[IN] Preview data callback parameters. For details, see the [NET_EHOME_PREVIEW_DATA_CB_PARAM](#) structure.

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is

returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ESTREAM_SetStandardPreviewDataCB

Register the preview (RTP stream) callback function.

Interface definition

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

parameter

iHandle

[IN] Preview listener handle, returned by [NET_ESTREAM_StartListenPreview](#).

pStruCBParam

[IN] The preview parameters to be called back. For details, see [NET_EHOME_PREVIEW_DATA_CB_PARAM](#).

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is

returned, call [NET_ESTREAM_GetLastError](#) to get the error code.

14.3.3 Playback Interface

NET_ESTREAM_SetPlayBackDataCB

Register a callback function to receive the playback stream.

Interface definition

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

parameter

iPlayBackLinkHandle

[IN] Playback handle, returned by the callback function *PLAYBACK_NEWINLINK_CB* registered by *NET_ESTREAM_StartListenPlayBack*.

pDataCBParam

[IN] Playback data callback parameters. For details, see the *NET_EHOME_PLAYBACK_DATA_CB_PARAM* structure.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

NET_ESTREAM_StartListenPlayBack

Enable playback monitoring of the streaming media server (SMS) and register a callback function to receive device connection requests.

Interface definition

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

parameter

pListenParam

[IN] Playback listener parameters. For details, see the *NET_EHOME_PLAYBACK_LISTEN_PARAM* structure.

Return

value -1 indicates failure, and other values indicate the handle of *NET_ESTREAM_StopListenPlayBack*. If -1 is returned, call *NET_ESTREAM_GetLastError* to get the error code.

NET_ESTREAM_StopListenPlayBack

Stops the playback monitoring of the streaming media server (SMS) and disconnects from the device.

Interface definition

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

parameter

iPlaybackListenHandle

[IN] The playback listener handle, returned by [NET_ESTREAM_StartListenPlayBack](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If
FALSE is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Note:

After calling this interface, all links of the listening port will be disconnected.

NET_ESTREAM_StopPlayBack

Stop forwarding playback stream of a certain channel.

Interface definition

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

parameter

iPlayBackLinkHandle

[IN] Playback listener handle, returned by the callback function [PLAYBACK_NEWLINK_CB](#) registered
by [NET_ESTREAM_StartListenPlayBack](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If
FALSE is returned, call [NET_ECMS_GetLastError](#) to get the error code.

Note:

For a listening port associated with multiple channels, after calling this interface, playback of only one of the channels will stop.

14.3.4 Voice Intercom Interface**NET_ESTREAM_StartListenVoiceTalk**

Enables voice intercom monitoring of the streaming media server (SMS) and registers a callback function to receive device connection requests.

Interface definition

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

parameter

pListenParam

[IN] Voice talk monitoring parameters. For details, see the [NET_EHOME_LISTEN_VOICETALK_CFG structure](#).

Return

value -1 indicates failure, and other values indicate the handle of [NET_ESTREAM_StopListenVoiceTalk](#).

If -1 is returned, call [NET_ESTREAM_GetLastError](#) to get the error code.

See also

[NET_ESTREAM_SendVoiceTalkData](#)**NET_ESTREAM_SetVoiceTalkDataCB**

Register a callback function to receive the audio stream.

Interface definition

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

parameter

IHandle

[IN] Voice talk handle, returned by the callback function [VOICETALK_NEWLINK_CB](#) registered by [NET_ESTREAM_StartListenVoiceTalk](#).

pStruCBParam

[IN] Voice intercom data callback parameters. For details, see the [NET_EHOME_VOICETALK_DATA_CB_PARAM](#) structure .

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is

returned, call [NET_ECMS_GetLastError](#) to get the error code.

NET_ESTREAM_SendVoiceTalkData

Forwards audio data to the device.

Interface definition

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

parameter

IHandle

[IN] Voice talk handle, returned by the [listening callback function \(VOICETALK_NEWLINK_CB \)](#) registered by
[NET_ESTREAM_StartListenVoiceTalk](#).

pVoicTalkData

[IN] Audio data to be forwarded. For details, see the [NET_EHOME_VOICETALK_DATA](#) structure .

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is

returned, call [NET_ECMS_GetLastError](#) to get the error code.

Notes •

The encoding type of the audio data to be forwarded to the device is the same as the voice intercom encoding type of the device itself. • For audio encoded using the G.722 encoding type, the data size that can be forwarded each time is 80 bytes; for audio encoded using the G.711 encoding type, the data size that can be forwarded each time is 160 bytes.

NET_ESTREAM_StopListenVoiceTalk

Stops the SMS voice intercom monitoring and disconnects the device.

Interface definition

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

parameter

IListenHandle

[IN] The listen handle of the voice talk, which is returned by *NET_ESTREAM_StartListenVoiceTalk*.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

Note:

After calling this interface, all links of the listening port will be disconnected.

NET_ESTREAM_StopVoiceTalk

Stop forwarding audio data of voice intercom on a certain channel.

Interface definition

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

parameter

IHandle

[IN] The listening handle of the voice talk, which is returned by the callback function *VOICETALK_NEWSLINK_CB* registered by *NET_ESTREAM_StartListenVoiceTalk*.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ECMS_GetLastError* to get the error code.

Note:

For a listening port associated with multiple channels, after calling this interface, the voice intercom of only one of the channels will stop.

14.4 Storage Management Module

14.4.1 Basic Interface

NET_ESS_Init

Initialize the storage management module library.

Interface definition

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

Return value

TRUE indicates success, *FALSE* indicates failure.

Remark

- If initialization fails, please check the path of loading library files and system environment. • This interface must be called in conjunction with *NET_ESS_Fini*.

NET_ESS_Init_V11

Set initialization parameters.

Interface definition

```
BOOL CALLBACK NET_ESS_Init_V11(   
    NET_EHOME_SS_INIT_PARAM *pParam );
```

parameter

pParam

For initialization parameters, see the *NET_EHOME_SS_INIT_PARAM* structure.

Return value

TRUE indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ESS_GetLastError* to get the error code.

NET_ESS_Finished

Deinitialize the storage management library and release the resources occupied by the storage server (SS).

Interface definition

```
BOOL CALLBACK NET_ESS_Fini( );
```

Return

value *TRUE* indicates success, *FALSE* indicates failure.

Note:

This interface must be called together with *NET_ESS_Init*.

NET_ESS_GENERATE_SECRETKEY

Calculate the encryption key.

Interface definition

```
BOOL CALLBACK NET_ESS_GENERATE_SECRETKEY( const
    char *pSrc, const char
    *pKey, *pSecretKeyOut,
    char
    DWORD dwSecretKeyLen )
```

parameter

pSrc

[IN] Input data buffer.

pKey

[IN] Key buffer.

pSecretKeyOut

[IN/OUT] Encryption key buffer.

dwSecretKeyLen

[IN] Encryption key buffer size.

Return

value *TRUE* indicates success, *FALSE* indicates failure.

If *FALSE* is returned, call [NET_ESS_GetLastError](#) to get the error code.

NET_ESS_HAMSHA256

Calculates the HMAC-SHA256 value.

Interface definition

```
BOOL CALLBACK NET_ESS_HAMSHA256( const
    char* pSrc, const char*
    pSecretKey, char* pSignatureOut,
    DWORD dwSignatureLen )
```

parameter

pSrc

[IN] Input data, usually pointing to the device ID.

pSecretKey

[IN] Key, usually points to EHomeKey.

pSignatureOut

[IN/OUT] Destination data buffer.

dwSignatureLen

[IN] Destination data buffer length.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call [NET_ESS_GetLastError](#) to get the error code.

NET_ESS_SetSDKInitCfg

Set storage server (SS) initialization parameters.

Interface definition

```
BOOL CALLBACK NET_ESS_SetSDKInitCfg(
    NET_EHOME_SS_INIT_CFG_TYPE enumType, void*
    const );
```

parameter

enumType

[IN] Initialization configuration type, see enumeration [NET_EHOME_SS_INIT_CFG_TYPE](#) for details .

nB

[IN] Initialize configuration parameters, depending on the configuration type (enumType), the corresponding relationship is as follows.

enumType	nB	illustrate
NET_EHOME_SS_INIT_CFG_SDK_PATH	<u>NET_EHOME_SS_LOCAL_SDK_PATH</u>	Set the SS component loading path (supported only by Linux version).
NET_EHOME_SS_INIT_CFG_CLOUD_TIME_DIFF	Points to a DWORD type, the value is to set the SS module, cloud storage request time difference, unit: minutes. Time difference value. Default 15 minutes, minimum 5 minutes, maximum 60 minutes.	
NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT	<u>NET_EHOME_IPADDRESS</u> sets the public network address of the SS module (when Used when there is intranet and extranet mapping).	
NET_EHOME_SS_INIT_CFG_LIBEAY_PATH	OpenSSL encryption library path (including library name). Example: "./libeay32.dll".	Set the path of the OpenSSL encryption library (libeay32.dll in Windows and libcrypto.so in Linux) used by the SS module.
NET_EHOME_SS_INIT_CFG_SSLEAY_PATH	OpenSSL communication library path (including library name). Example: "./ssleay32.dll".	Set the path of the OpenSSL communication library (ssleay32.dll in Windows and libssl.so in Linux) used by the SS module.

Return value

TRUE indicates success, *FALSE* indicates failure.

Note : •

This interface must be called before initialization, otherwise the settings will not take effect. • If

the call fails, the error code cannot be obtained or the correct error code cannot be obtained, because the error code must be initialized before it can be used.

Obtain.

• This interface does not check the validity of the OpenSSL path. The path is verified when the library is loaded. If the path is incorrect, other

The relevant interfaces will not report errors, but errors will be written to the log.

- When multiple modules are used in the same process, it is recommended to set the same OpenSSL path for all modules. For example, if the Registration Module (CMS) and the Alarm Management Module (AMS) are used in the same process, if the CMS has OpenSSL loaded, but the OpenSSL path of AMS is different from that of CMS, and the version number of the library is also different, AMS will use the loaded OpenSSL.
- Different modules (AMS/CMS/SS/SMS) are located in the same process. When loading the OpenSSL library, ensure that different modules use The OpenSSL library path used is valid, and the OpenSSL library version and library name are the same.

NET_ESS_SetLogFile

Set the parameters to enable the Storage Server (SS) log function.

Interface definition

```
BOOL CALLBACK NET_ESS_SetLogFile( iLogLevel,
    LONG      const char
    *strLogDir, bAutoDel
    BOOL
);
```

parameter

iLogLevel

[IN] Log type: 1-error log, 2-debug log, 3-information log.

strLogDir

[IN] Log file save path. The default save path in Windows operating system is "C:\\SdkLog\\\"", and the default save path in Linux operating system is "/home/sdklog/". If you want to use the default save path, set this parameter to "NULL".

bAutoDel

[IN] Whether to automatically delete the log file after a certain period of time: "TRUE" - yes (default), "FALSE" - no.

When "FALSE", log compression will be enabled. Every time 50 log files are generated, log compression will be performed (the 50 generated log files will be compressed into one compressed file).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ESS_GetLastError* to get the error code.

Note •

The save path of the log file must be an absolute path and must end with "\\", for example: "C:\\SdkLog\\\". • If the save path is changed, the changed path will be used to save the next log file.

- When the value of **bAutoDel** is "TRUE", the first log file will not be deleted because it contains the start time.
- Allow HCISUPSDK to modify log parameters during runtime. You can add the HCISUPSDK_Log_Switch.xml file in the runtime directory to configure the runtime log parameters.

[Log parameter configuration](#)

NET_ESS_GetBuildVersion

Get the version information of the storage server (SS).

Interface definition

```
DWORD CALLBACK NET_ESS_GetBuildVersion( );
```

The return

value returns the version number. The upper two bytes represent the major version, and the lower two bytes represent the minor version, for example: 0x02040001 (version 2.4.0.1).

NET_ESS_GetLastError

If the call fails or completes, you can get the error code.

Interface definition

```
DWORD CALLBACK NET_ESS_GetLastError( );
```

Return value

The return value is an error code. For details, see [ISUPSDK error code](#)

NET_ESS_GenerateStorageUri

Generates URL information for storage resources (images, etc.).

Interface definition

```
BOOL CALLBACK NET_ESS_GenerateStorageUri(
    NET_EHOME_SS_STORAGE_URI* pSSStorageUri );
```

parameter

pSSStorageUri

The URI information of the storage resource points to a [NET_EHOME_SS_STORAGE_URI](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ESS_GetLastError](#) to get the error code.

14.4.2 Listening Interface

NET_ESS_StartListen

Enables the storage server (SS) to listen and registers a callback function to receive stored files.

Interface definition

```
LONG CALLBACK NET_ESS_StartListen(  
    NET_EHOME_SS_LISTEN_PARAM *pSSListenParam );
```

parameter

pSSListenParam

[IN] Listening parameters. For details, see the [NET_EHOME_SS_LISTEN_PARAM](#) structure .

Return

value -1 indicates failure, and other values indicate the [handle](#) parameter of [NET_ESS_StopListen](#). If -1 is returned, call [NET_ESS_GetLastError](#) to get the error code.

Remarks

Before calling this interface, you need to call [interface Nnm](#) to set the HTTPS listening parameters; if HTTPS is not enabled, you need to call [interface Nnm](#) to clear the parameter [values](#), otherwise the set HTTPS listening parameters will continue to take effect.

NET_ESS_StopListen

Stop the storage server (SS) from listening.

Interface definition

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

parameter

iHandle

[IN] Listener handle, returned by [NET_ESS_StartListen](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ESS_GetLastError](#) to get the error code.

No

Set HTTPS listening parameters.

Interface definition

```
BOOL CALLBACK Nnrm(  
    NET_EHOME_SS_LISTEN_HTTPS_PARAM *m )
```

parameter

m

HTTPS listening parameters, see the structure [NET_EHOME_SS_LISTEN_HTTPS_PARAM](#) for details.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call [NET_ESS_GetLastError](#) to get the error code.

14.4.3 Client-related interfaces

NET_ESS_CreateClient

Create a client to upload files.

Interface definition

```
LONG CALLBACK NET_ESS_CreateClient(  
    NET_EHOME_SS_CLIENT_PARAM *pClientParam );
```

parameter

pClientParam

[IN] Client parameters. For details, see the [NET_EHOME_SS_CLIENT_PARAM](#) structure.

Return

value The return value is of LONG type. When it is less than 0, it indicates failure; when it is greater than or equal to 0, it indicates success. It is the client handle and is used for [NET_ESS_ClientSetTimeout](#), [NET_ESS_ClientSetParam](#), [NET_ESS_ClientDoUpload](#), [Parameters related to NET_ESS_ClientDoDownload](#) and [NET_ESS_DestroyClient](#) interfaces.

NET_ESS_ClientSetParam

Set client parameters for uploading files.

Interface definition

```
BOOL CALLBACK NET_ESS_ClientSetParam(
    LONG lHandle, const char
    *strParamName, const char
    *strParamVal );
```

parameter

lHandle

[IN] Client handle, returned by [NET_ESS_CreateClient](#).

strParamName

[IN] Client parameter name. For details, see the table below.

strParamName	describe
SS_CLIENT_FILE_PATH_PARAM_NAME	The path where the file is saved.
SS_CLIENT_VRB_FILENAME_CODE	VRB Protocol FilenameCode
SS_CLIENT_KMS_USER_NAME	Key Management Server (KMS) username
SS_CLIENT_KMS_PASSWORD	The password for the Key Management Server (KMS)
SS_CLIENT_CLOUD_AK_NAME	AccessKey for ISUP version 5.0
SS_CLIENT_CLOUD_SK_NAME	ISUP version 5.0 secret key (SecretKey)

strParamSelect

[IN] Client parameter value, determined by the parameter name (strParamName).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ESS_GetLastError* to get the error code.

NET_ESS_ClientSetTimeout

Set the timeout for uploading or downloading files through the client.

Interface definition

```
BOOL CALLBACK NET_ESS_ClientSetTimeout(  
    LONG IHandle,  
    DWORD dwSendTimeout,  
    DWORD dwRecvTimeout );
```

parameter

IHandle

[IN] Client handle, returned by *NET_ESS_CreateClient*.

dwSendTimeout

[IN] File upload timeout.

dwRecvTimeout

[IN] File download timeout.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ESS_GetLastError* to get the error code.

NET_ESS_ClientDoUpload

Start uploading files through the client.

Interface definition

```
BOOL CALLBACK NET_ESS_ClientDoUpload(  
    LONG IHandle, char  
    *strUrl,
```

```
LONG dwUrlLen );
```

parameter

IHandle

[IN] Client handle, returned by [NET_ESS_CreateClient](#).

strUrl

[IN] File URL.

dwUrlLen

[IN] File URL length.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call [NET_ESS_GetLastError](#) to get the error code.

NCnB

The client uploads resources (pictures) through memory data.

Interface definition

```
BOOL CALLBACK NCnBr(
    LONG IHandle,
    char* strUrl,
    DWORD dwUrlLen,
    void* pFileContent,
    DWORD dwContentLen
)
```

parameter

IHandle

[IN] Client handle, which is the return value of [NET_ESS_CreateClient](#).

strUrl

[IN/OUT] URL buffer.

dwUrlLen

[IN] URL buffer size.

pFileContent

[IN] Pointer to a resource buffer (such as image data).

dwContentLen

[IN] Resource buffer size.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ESS_GetLastError* to get the error code.

NET_ESS_ClientDoDelete

Delete the specified file from the storage server through the client.

Interface definition

```
BOOL NET_ESS_ClientDoDelete(  
    LONG IHandle, const  
    char *strUrl );
```

parameter**IHandle**

[IN] Client handle, returned by *NET_ESS_CreateClient*.

strUrl

[IN] URL of the specified file to be deleted, returned by *NET_ESS_ClientDoUpload*.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If *FALSE* is returned, call *NET_ESS_GetLastError* to get the error code.

NET_ESS_ClientDoDownload

Start downloading the file through the client.

Interface definition

```
BOOL CALLBACK NET_ESS_ClientDoDownload(  
    LONG IHandle, char  
    *strUrl, void  
    **pFileContent,  
    DWORD& dwContentLen );
```

parameter

IHandle

[IN] Client handle, returned by [NET_ESS_CreateClient](#).

strUrl

[IN] Image URL.

pFileContent

[OUT] Buffer for receiving downloaded data.

dwContentLen

[OUT] The size of the data in the receive buffer.

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call [NET_ESS_GetLastError](#) to get the error code.

NET_ESS_DestroyClient

Destroy the client.

Interface definition

```
BOOL CALLBACK NET_ESS_DestroyClient(  
    LONG IHandle );
```

parameter

IHandle

[IN] Client handle, returned by [NET_ESS_CreateClient](#).

Return

value *TRUE* indicates success, *FALSE* indicates failure. If

FALSE is returned, call [NET_ESS_GetLastError](#) to get the error code.

Chapter 15 Callback Functions

15.1 ASYNC_RESPONSE_CB

Response callback function

Callback function definition

```
typedef BOOL(CALLBACK * ASYNC_RESPONSE_CB)(  
    NET_EHOME_ASYNC_RESP_CB_DATA *pData,  
    void *pUser );
```

parameter

pData

[OUT] The callback response data. For details, see the structure *NET_EHOME_ASYNC_RESP_CB_DATA*.

nUser

[OUT] User parameter pointer.

Related interfaces

NCMmCnRnCB

15.2 DEVICE_REGISTER_CB

Register a callback function.

Callback function definition

```
typedef BOOL (CALLBACK *DEVICE_REGISTER_CB)(  
    LONG lUserID,  
    DWORD dwDataType, void  
    *pB,  
    DWORD dwOutLen, void  
    *pNB,  
    DWORD dwInLen, void  
    *pUser );
```

parameter

lUserID

[OUT] User ID, assigned by the SDK.

dwDataType

[OUT] callback data type, see enumeration [NET_EHOME_REGISTER_TYPE](#) for details. For devices that support different versions of ISUP, the corresponding callback data type logic is different, and each data type needs to be called back in a certain order, see [Device registration message processing flow](#)

B

[OUT] Output buffer. Different callback data types (dwDataType) correspond to different output parameters.

dwOutLen

[OUT] Output buffer size.

nB

[IN] Input buffer. Different input parameters correspond to different callback data types (dwDataType).

dwInLen

[IN] Input buffer size.

navel

[OUT] User parameters.

Remarks

- **The relationship between the input parameter (nB), callback data type (dwDataType), and output parameter (B)**

The system is shown below.

dwDataType	nB	B
ENUM_DEV_ON	NET_EHOME_SERVER_INFO	NET_EHOME_DEV_REG_INFO
ENUM_DEV_OFF	none	none
ENUM_DEV_ADDRESS_ CHANGED	NULL	NET_EHOME_DEV_REG_INFO
ENUM_EHOME50_DEV_AUTH EHomeKey (Call NET_ECMS_GetDeviceSessionK —		NET_EHOME_DEV_REG_INFO
ENUM_EHOME50_DEV_ SESSIONKEY	none	NET_EHOME_DEV_REG_INFO

dwDataType	nB	B
ENUM_EHOME50_DEV_DAS_REQ	<u>JSON_DasInfo</u>	<u>NET_EHOME_DEV_REG_INFO</u>
ENUM_DEV_SESSIONKEY_REQ	<u>NET_EHOME_DEV_REG_INFO_V12</u>	none
ENUM_DEV_DAS_REREGISTER	<u>NET_EHOME_DEV_REG_INFO_V12</u>	<u>NET_EHOME_SERVER_INFO_V50</u>
ENUM_DEV_DAS_PINGREO	<u>NET_EHOME_DEV_REG_INFO_V12</u>	none
ENUM_DEV_DAS_EHOMEKEY_ERROR	none	none
ENUM_DEV_SESSIONKEY_ERROR	<u>NET_EHOME_DEV_REG_INFO_Non</u> <u>V12</u>	
ENUM_DEV_SLEEP	none	none

- When calling back the device offline information, you must call NET_ECMS_ForceLogout to log out.

15.3 ECMSCallback

Asynchronous callback function of the Central Management Server (CMS)

Callback function definition

```
typedef void(CALLBACK *ECMCallback)(  
    NET_EHOME_CMSCB_DATA *pData,  
    *pUser  
);
```

parameter

pData

[OUT] The data to be called back. For details, see the structure NET_EHOME_CMSCB_DATA.

nUser

[OUT] User parameter pointer.

15.4 EHomeMsgCallBack

Alarm callback function.

Callback function definition

```
typedef BOOL (CALLBACK *EHomeMsgCallBack)( iHandle,
    LONG
    NET_EHOME_ALARM_MSG *pAlarmMsg, *pUser void

);
```

parameter

iHandle

[OUT] Alarm monitoring handle.

pAlarmMsg

[OUT] Alarm information. For details, see the structure [NET_EHOME_ALARM_MSG](#).

nUser

[OUT] User parameters.

Related interfaces

[NET_EALARM_StartListen](#)

15.5 EHomeSSMsgCallBack

File information callback function

Callback function definition

```
typedef BOOL (CALLBACK *EHomeSSMsgCallBack)( iHandle,
    LONG
    NET_EHOME_SS_MSG_TYPE enumType, void *B,
    dwOutLen,
    DWORD
    void *nB, dwInLen, *pUser
    DWORD

    void );
```

parameter

iHandle

[OUT] Handle.

enumType

[IN] Callback data type. For details, see the enumeration definition [NET_EHOME_SS_MSG_TYPE](#).

B

[OUT] Buffer for storing output parameters, depending on the type of data being called back (enumType).

dwOutLen

[OUT] Output buffer size.

nB

[IN] Buffer to store input parameters, depending on the type of data being called back (enumType).

dwInLen

[IN] Input buffer size.

navel

[OUT] User parameters.

Related interfaces

[NET_ECMS_StartListen](#)

Remark

When the storage service is enabled, this callback function (EHomeSSMsgCallBack) is required.

15.6 EHomeSSRWCallBack

Read and write callback functions

Callback function definition

```
typedef BOOL(CALLBACK *EHomeSSRWCallBack)
    LONG      iHandle,
    BYTE      byAct,
    const char *pFileName, void
    *pFileBuf, *dwFileLen,
    LONG      const char
    *pFileUrl, void *pUser

);
```

parameter

iHandle

[OUT] Handle.

byAct

[OUT] Operation type: 0-write, 1-read, 2-delete.

pFileName

[OUT] File name.

pFileBuf

[IN][OUT] Buffer where the file is saved.

dwFileLen

[IN][OUT] File size.

pFileUrl

[OUT] File URL.

navel

[OUT] User parameters.

Related interfaces

[NET_ECMS_StartListen](#)

15.7 EHomeSSRWCallBackEx

Read and write callback function extension

Callback function definition

```
typedef BOOL(CALLBACK *EHomeSSRWCallBackEx)(  
    LONG iHandle,  
    NET_EHOME_SS_RW_PARAM* pRwParam,  
    NET_EHOME_SS_EX_PARAM* pExStruct );
```

parameter

iHandle

Storage component server listening handle, [the return value of NET_ESS_StartListen](#).

pRwParam

Used to pass common fields of four storage protocols, [see NET_EHOME_SS_RW_PARAM](#).

pExStruct

Used to pass storage protocol specific fields, see [*NET_EHOME_SS_EX_PARAM*](#).

15.8 EHomeSSStorageCallBack

Storage information callback function

Callback function definition

```
typedef BOOL (CALLBACK *EHomeSSStorageCallBack)(  
    LONG iHandle, const  
    char* *pFileName, void *pFileBuf,  
  
    DWORD dwFileLen, char  
    *pFilePath, void *pUser );
```

parameter

iHandle

[OUT] Storage handle.

pFileName

[OUT] File name.

pFileBuf

[OUT] The size of the buffer for saving files.

pFilePath

[OUT] File save path.

dwInLen

[IN] Input buffer size.

nave1

[OUT] User parameters.

Related interfaces

[*NET_ECMS_StartListen*](#)

15.9 xcnCBc

Callback function that receives exception information

Callback function definition

```
typedef void(CALLBACK xctinCBc)(  
    DWORD dwType,  
    int iUserID,  
    int iHandle,  
    void *pUser  
)
```

parameter

dwType

[OUT] Exception information type, details are as follows:

Macro Definition	value	describe
EHOME_VOICETALK_EXCEPTION	0x101	The voice intercom is abnormal.
EHOME_PREVIEW_EXCEPTION	0x102	Preview exception.
EHOME_PLAYBACK_EXCEPTION	0x103	Playback exception.
EHOME_AUDIOTALK_EXCEPTION	0x104	The audio stream is abnormal.
EHOME_CMSALARM_EXCEPTION	0x105	Alarm reception abnormality.

iUserID

[OUT] User ID.

iHandle

[OUT] Handle (session ID).

nUser

[OUT] User parameter pointer.

15.10 fVoiceDataCallBack

Audio data callback function

Interface definition

```
typedef void (CALLBACK *fVoiceDataCallBack)(  
    LONG VoiceComHandle,  
    char *RcvB,  
    int dwBufSize,
```

```
int dwEncodeType,
BYTE byAudioFlag, void
*pUser );
```

parameter

VoiceComHandle

[OUT] Voice intercom or audio forwarding handle, returned by *NET_ECMS_StartVoiceTalk*.

RcvB

[OUT] Pointer to the buffer where the audio file is stored.

dwBufSize

[OUT] Audio data size.

dwEncodeType

[OUT] Audio data encoding type: 0-G.722, 1-G.711U, 2-G.711A, 3-G.726, 4-AAC, 5-MP2L2, 6-PCM

byAudioFlag

[OUT] Audio data source. 0-local acquisition, 1-from device.

navel

[OUT] User parameters.

15.11 nCnCBc

Long connection callback function

Interface definition

```
typedef BOOL(CALLBACK nCnCBc)(
    LONG iHandle,
    LONG_LINK_MSG enMsg, void
    *B,
    DWORD dwOutLen, void
    *pUser );
```

parameter

iHandle

[OUT] Long connection handle, returned by *NCMnCnC*.

enMsg

[OUT] Long connection information type, enumerated in *LONG_LINK_MSG*.

B

[OUT] Output buffer.

dwOutLen

[OUT] Data size in the output buffer.

nUser

[OUT] User pointer.

15.12 PASSTHROUGHDATA CALLBACK

Data transmission callback function

Interface definition

```
typedef void (CALLBACK* PASSTHROUGHDATA CALLBACK)(  
    DWORD dwProxyType,  
    LONG iListenHandle, void  
    pDeviceID,  
    DWORD dwDevIDLen, void B,  
  
    DWORD dwDataLen, void  
    pUser );
```

parameter

dwProxyType

[OUT] Proxy type. The default type is "ENUM_PROXY_TYPE_NETSDK" (private protocol proxy).

iListenHandle

[OUT] Proxy listener handle, returned by *NET_ECMS_StartListenProxy*.

pDeviceID

[OUT] Device ID.

dwDevIDLen

[OUT] Device ID length.

B

[OUT] Output buffer.

dwDataLen

[OUT] Output buffer size.

navel

[OUT] User pointer.

15.13 PLAYBACK_DATA_CB

Callback function for replaying data

Interface definition

```
typedef BOOL (CALLBACK *PLAYBACK_DATA_CB)(  
    LONG iPlayBackLinkHandle,  
    NET_EHOME_PLAYBACK_DATA_CB_INFO *pDataCBInfo, void  
    *pUserData );
```

parameter

iPlayBackLinkHandle

[OUT] Playback handle, returned by *PLAYBACK_NEWLINK_CB*.

pDataCBInfo

[OUT] The playback data being called back. For details, see *NET_EHOME_PLAYBACK_DATA_CB_INFO*.

pUserData

[OUT] User parameters.

Return value

Returning *TRUE* means starting the playback of the channel, and returning *FALSE* means canceling the playback of the channel and automatically releasing resources.

15.14 PLAYBACK_NEWLINK_CB

Callback function for playback request

Callback function definition

```
typedef BOOL (CALLBACK *PLAYBACK_NEWLINK_CB)  
    ( iPlayBackLinkHandle,  
    NET_EHOME_PLAYBACK_NEWLINK_CB_INFO *pNewLinkCBInfo, void  
    *pUserData );
```

parameter

iPlayBackLinkHandle

[OUT] Playback handle.

pNewLinkCBInfo

[OUT] The callback playback request. For details, see the [NET_EHOME_PLAYBACK_NEWLINK_CB_INFO](#) structure .

pUserData

[OUT] User parameters.

Return value

Returning *TRUE* means starting the playback of the channel, and returning *FALSE* means canceling the playback of the channel and automatically releasing resources.

Related interfaces

[NET_ESTREAM_StartListenPlayBack](#)

15.15 PREVIEW_DATA_CB

Callback function for previewing data

Callback function definition

```
typedef BOOL (CALLBACK *PREVIEW_DATA_CB)(  
    LONG iPreviewHandle,  
    NET_EHOME_PREVIEW_CB_MSG *pPreviewCBMsg, void  
    *pUserData );
```

parameter

iPreviewHandle

[OUT] Preview handle, returned by [PREVIEW_NEWLINK_CB](#) .

pPreviewCBMsg

[OUT] The preview data being called back, see the structure [NET_EHOME_PREVIEW_CB_MSG](#) for details

pUserData

[OUT] User parameters.

Return Value

Returning *TRUE* means starting the preview of the channel, and returning *FALSE* means canceling the preview of the channel and automatically releasing resources.

member	type of data	describe
byStartTime	char[]	Start time, format: YYYY-MM-DD HH:MM:SS (year-month-day hour:minute:second). The maximum length is 32 bytes (corresponding to macro defined as "MAX_TIME_LEN").
byStopTime	char[]	End time, format: YYYY-MM-DD HH:MM:SS (year-month-day hour:minute:second). The maximum length is 32 bytes (corresponding to macro defined as "MAX_TIME_LEN").
struHeatmapValue	<u>NET_EHOME_HEATMA</u> <u>P_VALUE</u>	Heat value.
struPixelArraySize	<u>NET_EHOME_PIXEL_A</u> <u>RRAY_SIZE</u>	Heatmap data size.
byPixelArrayData	BYTE[]	Heatmap data number. The maximum length is 512 bytes.
byRetransFlag	BYTE	Retransmission flag: 0-real-time packet, 1-retransmission packet.
I would	BYTE	The difference between the start/end time and UTC time (hours), value It is -12, -10, ..., +12, +14, and 0x indicates invalid.
bymM BYTE		The difference between the start/end time and UTC time (minutes), value -30, 0, 30, 45, 0x indicates invalid.
byRes	BYTE[]	Reserved. The maximum length is 61 bytes.

A.1.64 NET_EHOME_HEATMAP_VALUE

Heat value information structure

member	type of data	describe
dwMaxHeatMapValue DWORD		Maximum heat value.
dwMinHeatMapValue DWORD		Minimum heat value.
dwTimeHeatMapValue It is	DWORD	Average heat value.

See also

[NET_EHOME_HEATMAP_REPORT](#)

A.1.65 NET_EHOME_HTTP_PARAM

About the structure of **HTTP** page transmission

member	type of data	describe
nB	void*	Input buffer (XML format data).
dwInSize	DWORD	Enter the buffer size.
B	void*	Output buffer (XML format data).
dwOutSize	DWORD	Output buffer size.
dwReturnedXMLLen DWORD		The length of the XML data actually received from the device.
dwRecvTimeOut	DWORD	Accept timeout in milliseconds.
byRes	BYTE[]	Reserved. The maximum length is 28 bytes.

A.1.66 NET_EHOME_HUMAN_FEATURE

ISUPSDK (General) Development Guide

Human attribute information structure

member	type of data	describe
byAgeGroup	BYTE	Age group: 1-infants, 2-children, 3-teenagers, 4-young adults Years, 5-young, 6-middle-aged, 7-middle-aged, 8-middle-aged, 9-elderly.
bySex	BYTE	Gender: 1-Male, 2-Female.
byEyeGlass	BYTE	Do you wear glasses: 1-No, 2-Yes.
cityMask	BYTE	Whether to wear a mask, 1-not wearing, 2-wearing.
byRes	BYTE[]	Reserved. The maximum length is 12 bytes.

A.1.67 NET_EHOME_IDENTIFICATION

Terminal ID structure

member	type of data	describe
bySerialNumber	BYTE[]	Serial number, currently not supported. Maximum length is 32 bytes.
upC	BYTE[]	Terminal ID. The maximum length is 32 bytes.

A.1.68 NET_EHOME_IMAGE_CFG

Image quality parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byHue	BYTE	Hue, value range: [0,255].
byContrast	BYTE	Contrast, value range: [0,255].
byBright	BYTE	Brightness, value range: [0,255].

member	type of data	describe
the village of BYTE		Saturation, value range: [0,255].
byRes	BYTE[]	Reserved, set to 0. The maximum length is 24 bytes.

A.1.69 NET_EHOME_INSERT_CHARACTER

About the structure of interstitial text information

member	type of data	describe
dwMessageCnt	DWORD	Insert message content.
struTextMessage	<u>NET_EHOME_TEXT_M</u> <u>TEST[]</u>	Inserted text message. Maximum length is 5 bytes (corresponding to macro Defined as "NET_EHOME_MAX_TEXT_MESSAGE_ ON ONE").
szMsgPos	BYTE	Display position of text messages: top, middle, bottom, The maximum length is 32 bytes (the corresponding macro is defined as "NET_EHOME_MAX_TYPE_LEN").
n	<u>NET_EHOME_POSITION</u> <u>N_INFO</u>	Custom location for text messages. This member is only The value of szMsgPos is valid only when it is customized.
Cc <u>NET_EHOME_CHAR_EF</u> <u>FECT</u>		Character effects.

A.1.70 NET_EHOME_INSERT_EFFECT

ISUPSDK (General) Development Guide

Insert effect structure

member	type of data	describe
Cc <i>NET_EHOME_CHAR_EF</i> <u>FECT</u>	_____	Character effects.
dwPageTime	DWORD	Page display time.
dwScrollSpeedWeb DWORD		Screen scrolling speed.

See also

[NET_EHOME_INSERT_INFO_V20](#)

A.1.71 NET_EHOME_INSERT_INFO

Insert parameter structure

member	type of data	describe
dwInsertType	DWORD	Insertion type, see enumeration <u>NET_EHOME_PROGRAM_INSERT_TYPE</u>
struMaterialInfo	<u>NET_EHOME_INSERT</u> <u>TEXT_INFO</u>	Material information.
struProgramInfo	<u>NET_EHOME_INSERT</u> <u>TEXT_INFO</u>	Program information.
struPosInfo	<u>NET_EHOME_POSITION</u> <u>N_INFO</u>	location information.
nc <i>NET_EHOME_INSERT</i> <u>EFFECT</u>	_____	Insert effect.

A.1.72 NET_EHOME_INSERT_INFO_V20

ISUPSDK (General) Development Guide

Insert parameter structure (V20)

member	type of data	describe
dwInsertType	DWORD	Insertion type, see enumeration <u>NET_EHOME_PROGRAM_INSERT_TYPE</u>
struMaterialInfo	<u>NET_EHOME_INSERT</u> <u>TEXT_INFO</u>	Material information.
struProgramInfo	<u>NET_EHOME_INSERT</u> <u>TEXT_INFO</u>	Program information.
struPosInfo	<u>NET_EHOME_POSITION</u> <u>N_INFO</u>	location information.
nc NET_EHOME_INSERT_	<u>EFFECT</u>	Insert effect.
dwPosMode	DWORD	Coordinate type, see enumeration <u>NET_EHOME_POS_MODE</u>
byRes	BYTE[]	Reserved. The maximum length is 64 bytes.

A.1.73 NET_EHOME_INSERT_TEXT_INFO

Insert material or program information structure

member	type of data	describe
dwTextNo	DWORD	The serial number of the insert file, that is, the material ID or program ID.
szPlayMode	BYTE[]	Play mode: byTime (insert by time) or byEndTime (Insert according to the end time). The maximum length is 32 bytes.
dwCountNum	DWORD	Playback times, not supported yet.
white DWORD		play time.

member	type of data	describe
playEndTime	BYTE[]	Insert according to the end time. The maximum length is 32 characters Festival.
dwTextSeq	DWORD	Used to determine whether the file ID and the file correspond to each other.

A.1.74 NET_EHOME_IPADDR

IP address structure

member	type of data	describe
slpV4	char[]	Device IPv4 address. Maximum length is 16 bytes.
slpV6	char[]	Device IPv6 address. Maximum length is 128 bytes.

A.1.75 NET_EHOME_IPADDRESS

Address information structure

member	type of data	describe
sIP	char[]	IP address. The maximum length is 128 bytes.
wPort	WORD	The port number.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 2 bytes.

A.1.76 NET_EHOME_IPC_INFO

ISUPSDK (General) Development Guide

Network camera information structure

member	type of data	describe
enumIPCType	<i>NET_EHOME_IPC_TYPE</i>	network camera type.
dwId	DWORD	The ID of the network camera.
chanNum	DWORD	The channel number of the associated program.
enumAddressType	<i>NET_EHOME_ADDRESS_TYPE</i>	The address type.
szIpcState	BYTE[]	Network camera status: online, offline (line). The maximum length is 32 bytes.
szHostName	BYTE	The domain name of the network camera. The maximum length is 32 bytes.
szIpVersion	BYTE	IP address version: v4, v6. The maximum length is 16 bytes.
szIpAddress	BYTE	IPv4 address. Maximum length is 32 bytes.
szIpv6Address	BYTE	IPv6 address. Maximum length is 32 bytes.
dwPortNo	DWORD	The port number of the network camera.
dwIpcChanNum	DWORD	The channel number of the network camera.
szUserName	BYTE	The user name of the network camera. The maximum length is 52 bytes.
szPassWord	BYTE	The password of the network camera. The maximum length is 32 bytes.
enumTransmitProtocol	<i>NET_EHOME_TRANSMIT_PROTOCOL</i>	Transport protocol type.
enumStreamType	<i>NET_EHOME_STREAM_TYPE</i>	Stream type.

A.1.77 NET_EHOME_LINKAGE_ALARMOUT

Alarm output linkage parameter structure

member	type of data	describe
dwAnalogAlarmOutNu m	DWORD	Number of analog alarm outputs, read only.
byAnalogAlarmOut	BYTE[]	Analog alarm output status, represented by array subscript: 0-disabled 1-Enable. The maximum length is 32 bytes (corresponding to the macro definition is defined as "MAX_ANALOG_ALARMOUT").
byRes	BYTE[]	Reserved, set to 0. The maximum length is 5000 bytes.

A.1.78 NET_EHOME_LINKAGE_PTZ

PTZ linkage parameter structure

member	type of data	describe
byUsePreset	BYTE	Whether to call the preset point: 0-No, 1-Yes.
byUseCurise	BYTE	Whether to call cruise: 0-No, 1-Yes.
byUseTrack	BYTE	Whether to call the track: 0-No, 1-Yes.
byRes1	BYTE	reserve.
wPresetNo	WORD	Preset point number, ranging from 1 to 256. Some devices Supports 300 preset points.
wCuriseNo	WORD	Cruise number, ranging from 1 to 16.
wTrackNo	WORD	Track number, ranging from 1 to 16.
byRes2	BYTE[]	Reserved. The maximum length is 6 bytes.

A.1.79 NET_EHOME_LISTEN_PREVIEW_CFG

Preview monitoring parameter structure

member	type of data	describe
strIPAdress	<u>NET_EHOME_IPADDRE</u> SS	Local monitoring information. When the IP is 0.0.0.0, the default is the local address; when there are multiple network cards, the default is The first address obtained by the operating system.
fnNewLinkCB	<u>PREVIEW_NEWLINK_C</u> B	Callback function, used to receive the response message of the preview request.
navel	void*	User data.
byLinkMode	BYTE	Access method for monitoring requests: 0-TCP, 1-UDP.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 127 bytes.

Related interfaces

[NET_ESTREAM_StartListenPreview](#)

A.1.80 NET_EHOME_LISTEN_VOICETALK_CFG

Voice intercom monitoring parameter structure

member	type of data	describe
strIPAdress	<u>NET_EHOME_IPADDRE</u> SS	Local monitoring information. When the IP is 0.0.0.0, the default is the local address; when there are multiple network cards, the default is The first address obtained by the operating system.
fnNewLinkCB	<u>VOICETALK_NEWLINK_C</u> CB	Callback function for voice intercom request.
pUserData	void*	User parameters.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 128 bytes.

A.1.81 NET_EHOME_LOCAL_ACCESS_SECURITY

ISUPSDK (General) Development Guide

Security parameter structure for device access to server

member	type of data	describe
dwSize	DWORD	The size of the structure.
byAccessSecurity	BYTE	Security access level: 0-compatible mode (allows any ISUP Protocol version of the device access), 1-Normal mode (only supports ISUP4.0 (excluding) and below protocol versions) 2-Full mode (only supports devices with ISUP4.0 (inclusive) or higher protocol versions) equipment access).
byRes	BYTE[]	Reserved, set to 0. The maximum length is 127 bytes.

A.1.82 NET_EHOME_LOCAL_DEV_PINGREO

Regarding whether to call back the heartbeat structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byEnablePingReoCallback	BYTE	Whether to call back the heartbeat: 0-No (default), 1-Yes.
byRes	BYTE[]	Reserved. The maximum value length is 63 bytes.

A.1.83 NET_EHOME_LOCAL_GENERAL_CFG

Common parameter structure

member	type of data	describe
byAlarmPictureSepara the	BYTE	Whether to transmit alarm data and images separately: 0-No, 1-Yes. If separated, the alarm type returned in the callback function is "EHOME_ISAPI_ALARM"ý
byRes	BYTE[]	Reserved, set to 0. The maximum length is 127 bytes.

A.1.84 NET_EHOME_LOCAL_PLAYBACK_PARAM

Table A-8 Playback parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byPlayBackSync	BYTE	Whether to enable playback synchronous reception, 0-indicates using asynchronous method 1- indicates the use of synchronous mode.
byRes	BYTE[]	Reserved. Maximum length is 131 bytes.

A.1.85 NET_EHOME_LONG_CFG_INPUT

Long connection parameter structure

member	type of data	describe
fnDataCallBack	<u>__nCnCBc.long.connection.callback</u> function.	
navel	void*	User pointer.
byRes	BYTE[]	Reserved. The maximum length is 32 bytes.

A.1.86 NET_EHOME_LONG_CFG_SEND

Data structure sent via long connection

member	type of data	describe
B	void*	A buffer that holds data to be sent.
dwDataLen	DWORD	The size of the buffer for data to be sent.
byRes	BYTE[]	Reserved. The maximum length is 32 bytes.

A.1.87 NET_EHOME_MAKE_I_FRAME

Mandatory I frame parameter structure

member	type of data	describe
iChannel	int	Channel number.
dwStreamType	DWORD	Stream type: 0-main stream, 1-sub stream, 2-reserved.
byRes	BYTE[]	Reserved. The maximum length is 40 bytes.

A.1.88 NET_EHOME_MANUAL_IOOUT_CTRL

Alarm output manual control parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwChan	DWORD	Alarm output number, starting from 0.
dwDelayTime	DWORD	Alarm output duration (s), a value of 0 means the alarm is always output. out.
bycn	BYTE	Control type: 0-turn off alarm output, 1-turn on alarm output.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 19 bytes.

A.1.89 NET_EHOME_MPDATA

Vehicle-mounted equipment passenger flow statistics data structure

member	type of data	describe
byIndex	BYTE	Door number, starting from 1.
byVideoChannel	BYTE	Video channel number, starting from 1.
byRes	BYTE	reserve.
atLevel	BYTE	Crowding in the car: 0-empty (dwCount<20), 1-normal (20<=dwCount<=30), 2-more (30<=dwCount<=50), 3-crowded (dwCcount>=50)
I would	char[]	Statistics start time/door opening time, format: YYYY-MM-DD HH:MM:SS (year-month-day hour:minute:second). The maximum length is 32 bytes (the corresponding macro is defined as "MAX_TIME_LEN").
I would	char[]	Stop statistics time/closing time, format: YYYY-MM-DD HH:MM:SS (year-month-day hour:minute:second). The maximum length is 32 bytes (the corresponding macro is defined as "MAX_TIME_LEN").
dwEnterNum	DWORD	The number of people who enter during the statistical time period.
dwLeaveNum	DWORD	The number of people who left during the statistical time.
dwCount	DWORD	The number of people in the car at the current moment.

A.1.90 NET_EHOME_MPGPS

GPS positioning information structure for vehicle passenger flow statistics

member	type of data	describe
iLongitude	LONG	Longitude information.
	LONG	Dimension information.

member	type of data	describe
iSpeed	LONG	speed.
cn	LONG	direction.

A.1.91 NET_EHOME_NETWORK_CFG

Network parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
struEtherNet	<u>NET_EHOME_ETHERNE</u> <i>T</i> —	Ethernet port information.
struGateWayIP	<u>NET_EHOME_IPADDR</u> gateway address.	
MC	<u>NET_EHOME_IPADDR</u> multicast address.	
struDDNSServer1IP <u>NET_EHOME_IPADDR</u>	The IP address of DDNS1.	
struDDNSServer2IP <u>NET_EHOME_IPADDR</u>	The IP address of DDNS2.	
struAlarmHostIP	<u>NET_EHOME_IPADDR</u> Alarm host IP address.	
wAlarmHostPort	WORD	Alarm host port number.
wIPResolverPort	WORD	Resolution server port number.
struIPResolver	<u>NET_EHOME_IPADDR</u> resolution server address.	
struPPPoE	<u>NET_EHOME_PPPOEFC</u> <i>G</i> —	PPPoE parameters.
wHTTPPort	WORD	HTTP communication port number.
byRes	BYTE[]	Reserved. The maximum length is 674 bytes.

A.1.92 NET_EHOME_NEWSLINK_CB_MSG

ISUPSDK (General) Development Guide

Preview request parameter structure (applicable to **64-bit Windows or Linux** operating systems)

member	type of data	describe
szDeviceID	BYTE[]	Device ID. The maximum length is 256 bytes (corresponding to the macro definition is MAX_DEVICE_ID_LEN).
iSessionID	LONG	Device streaming session ID.
dwChannelNo	DWORD	Channel number.
byStreamType	BYTE	Stream type: 0-main stream, 1-sub stream.
byRes1	BYTE[]	Reserved, the maximum length is 2 bytes.
byStreamFormat	BYTE	Code stream encapsulation format: 0-PS, 1-standard stream format.
sDeviceSerial	char[]	Device serial number. Maximum length is 12 bytes (corresponding to macro definition is defined as "NET_EHOME_SERIAL_LEN").
fnPreviewDataCB	<i>PREVIEW_DATA_CB</i> data callback	function, its size is 8 bytes.
pUserData	void*	User parameters, its size is 8 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 96 bytes.

Preview request parameter structure (applicable to **32-bit Windows or Linux** operating systems)

member	type of data	describe
szDeviceID	BYTE[]	Device ID. The maximum length is 256 bytes (corresponding to the macro definition is MAX_DEVICE_ID_LEN).
iSessionID	LONG	Device streaming session ID.
dwChannelNo	DWORD	Channel number.
byStreamType	BYTE	Stream type: 0-main stream, 1-sub stream.
byRes1	BYTE[]	Reserved, the maximum length is 2 bytes.
byStreamFormat	BYTE	Code stream encapsulation format: 0-PS, 1-standard stream format.

member	type of data	describe
sDeviceSerial	char[]	Device serial number. Maximum length is 12 bytes (corresponding to macro definition is defined as "NET_EHOME_SERIAL_LEN").
fnPreviewDataCB	<u>PREVIEW_DATA_CB</u> data callback	function, its size is 4 bytes.
byRes2	BYTE[]	Reserved, the maximum length is 4 bytes.
pUserData	void*	User parameters, its size is 4 bytes.
byRes3	BYTE[]	Reserved, the maximum length is 4 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 96 bytes.

A.1.93 NET_EHOME_NOTICE_PICURL

Image URL information structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byDeviceID	char[]	Device registration ID. Maximum length is 256 bytes (corresponding to macro is defined as "MAX_DEVICE_ID_LEN").
wPicType	WORD	Image type: 0-timed snapshot upload center, 1-alarm snapshot Upload center, 2-Manual capture upload center, 3-Download image 4-External device triggers the capture upload center.
wAlarmType	WORD	Alarm type: 6-alarm input, 7-video blocking alarm, 8-Motion detection alarm, 11-PIR alarm, 12-door magnetic alarm, 3-SOS alarm. This parameter is only available when wPicType is set to 1. The number is valid.
dwAlarmChan	DWORD	Alarm channel number. This is only available when wPicType is set to 1. The parameters are valid.

member	type of data	describe
byAlarmTime	char[]	Alarm trigger time, format: YYYY-MM-DD HH:MM:SS (year-month-day hour:minute:second). Only when wPicType is set to 1 This parameter is valid when the maximum length is 32 bytes (corresponding to the macro is defined as "MAX_TIME_LEN").
dwCaptureChan	DWORD	Snapshot channel number.
byPicTime	char[]	Image display time, format: YYYY-MM-DD HH:MM:SS (year-month-day hour:minute:second). The maximum length is 32 bytes (for should be defined as "MAX_TIME_LEN").
byPicUrl	char[]	Image URL. Maximum length is 512 bytes (corresponding to macro definition is MAX_URL_LEN).
dwManualSnapSeq DWORD		Request sequence number, only when PicType is set to 2 or 3. This parameter is valid.
byRetransFlag	BYTE	Retransmission flag: 0-real-time packet, 1-retransmission packet.
I would	BYTE	The difference between the start/end time and UTC time (hours), value It is -12, -10, ..., +12, +14, and 0x indicates invalid.
bymM BYTE		The difference between the start/end time and UTC time (minutes), value -30, 0, 30, 45, 0x indicates invalid.
byRes	BYTE[]	Reserved. The maximum length is 29 bytes.

A.1.94 NET_EHOME_NOTIFY_FAIL_INFO

ISUPSDK (General) Development Guide

Asynchronous failure notification information structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byDeviceID	char[]	Device registration ID. The maximum value length is 256 bytes (macro definition: "MAX_DEVICE_ID_LEN").
wFailedCommand WORD		Failed commands: 1-Capture failed.
wPicType	WORD	Image type: 0-timed capture and upload to the center, 1-alarm Trigger capture and upload to the center, 2-manual capture and upload to Center, 3-Download pictures, 4-External devices trigger capture and upload Passed to the center. Only if wFailedCommand is set to "1" This parameter is valid when
dwManualSnapSeq DWORD		Request serial number, valid when PicType is 2 or 3.
byRetransFlag	BYTE	Retransmission flag: 0-real-time packet, 1-retransmission packet.
byRes	BYTE[]	Reserved. The maximum value length is 31 bytes.

A.1.95 NET_EHOME_PASSTHROUGH_PARAM

Protocol parameter structure

member	type of data	describe
dwSequence	DWORD	Message sequence number.
dwUUID	DWORD	Session ID.
byRes	BYTE[]	Reserved. The maximum length is 64 bytes.

A.1.96 NET_EHOME_PASSWORD_CFG

Terminal password parameter structure

member	type of data	describe
szOldPwd	BYTE[]	Old password. Maximum length is 32 bytes.
szNewPwd	BYTE[]	New password. Maximum length is 32 bytes.

A.1.97 NET_EHOME_PIC_CFG

OSD display parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byChannelName	BYTE[]	Channel name. Maximum length is 32 bytes (corresponding to macro definition is NAME_LEN).
bIsShowChanName BYTE		Whether to display channel name: 0-No, 1-Yes.
wChanNameXPos	WORD	The X-axis coordinate of the channel name display is 704 × 576. Configuration, the coordinate value is a multiple of 16.
wChanNameYPos	WORD	The Y-axis coordinate of the channel name is displayed in 704 × 576. Configuration, the coordinate value is a multiple of 16.
bIsShowOSD	BYTE	Whether to display date information: 0-No, 1-Yes.
wOSDXPos	WORD	The X-axis coordinate of the OSD display is configured as 704 × 576. The coordinate value must be a multiple of 16.
wOSDYPos	WORD	The Y-axis coordinate of the OSD display is configured according to 704 × 576. The coordinate value must be a multiple of 16.
byOSDType	WORD	OSD format: year/month/day, 0: XXXX-XX-XX (year-month-day), 1: XX-XX-XXXX (month-day-year), 2: XXXX year XX Month XX day, 3: XX month XX day XXXX year, 4: XX-XX-XXXX(day-month-year), 5: XX day XX month XXXX year

member	type of data	describe
byOSDAtrib	BYTE	OSD properties: 0: Do not display OSD, 1: Transparent, flashing, 2: transparent, no flicker, 3: flicker, opaque, 4: no Transparent, no flickering
byRes1	BYTE[]	Reserved, set to 0. The maximum length is 2 bytes.
bIsShowWeek	BYTE	Whether to display the day of the week: 0-No, 1-Yes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 64 bytes.

A.1.98 NET_EHOME_PIC_FILE

Image file search result structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
szFileName	char[]	Image file name. Maximum length is 100 bytes (corresponding to macro definition is defined as "MAX_FILE_NAME_LEN").
struPicTime	<u>NET_EHOME_TIME</u> Image generation time.	
dwFileSize	DWORD	Image file size.
dwFileMainType	DWORD	Capture type: 0x all types, 0 (0x00) - scheduled capture Figure, 1 (0x01) - motion detection capture, 2 (0x02) - report Alarm capture, 3 (0x03) - Alarm or motion detection capture, 4 (0x04) - Alarm and motion detection capture, 5 (0x05) - Command Command trigger capture, 6 (0x06) - manual capture, 7 (0x07) - Vibration alarm capture, 8 (0x08) - Environmental alarm trigger capture Figure, 9 (0x09) - Intelligent alarm capture, 10 (0x0a) - PIR alarm capture, 11 (0x0b) - wireless alarm capture, 12 (0x0c)-call for help and capture of alarm, 13 (0x0d)-face detection Capture, 14 (0x0e) - Cross-border detection capture, 15 (0x0f) -

ISUPSDK (General) Development Guide

member	type of data	describe
		Intrusion area detection capture, 16 (0x10) - scene change detection Test capture, 17 (0x11) - device local playback screenshot, 18 (0x12)-Intelligent detection and capture, 19 (0x13)-Enter the area Detection snapshot, 20 (0x14) - Leaving area detection snapshot, 21 (0x15) - Wandering detection capture, 22 (0x16) - Personnel Gathering detection capture, 23 (0x17) - fast movement detection capture Figure, 24 (0x18) - Parking detection capture, 25 (0x19) - Item left behind detection capture, 26 (0x1a) - Item taken detection Capture, 27 (0x1b) - License plate detection capture, 28 (0x1c)-Client uploads picture
dwFileIndex	DWORD	Image file number, starting from 0.
l would	BYTE	The difference between the start/end time and UTC time (hours), value It is -12, -10, ..., +12, +14, and 0x indicates invalid.
bymM BYTE		The difference between the start/end time and UTC time (minutes), value -30, 0, 30, 45, 0x indicates invalid.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 126 bytes.

A.1.99 NET_EHOME_PIC_FILE_COND

Image file search condition structure

member	type of data	describe
dwChannel	DWORD	Channel number.
dwPicType	DWORD	Capture type: 0x all types, 0 (0x00) - scheduled capture Figure, 1 (0x01) - motion detection capture, 2 (0x02) - report Alarm capture, 3 (0x03) - Alarm or motion detection capture, 4 (0x04) - Alarm and motion detection capture, 5 (0x05) - Command Command trigger capture, 6 (0x06) - manual capture, 7 (0x07) -

ISUPSDK (General) Development Guide

member	type of data	describe
		<p>Vibration alarm capture, 8 (0x08) - Environmental alarm trigger capture</p> <p>Figure, 9 (0x09) - Intelligent alarm capture, 10 (0x0a) -</p> <p>PIR alarm capture, 11 (0x0b) - wireless alarm capture, 12 (0x0c)-call for help and capture of alarm, 13 (0x0d)-face detection</p> <p>Capture, 14 (0x0e) - Cross-border detection capture, 15 (0x0f) -</p> <p>Intrusion area detection capture, 16 (0x10) - scene change detection</p> <p>Test capture, 17 (0x11) - device local playback screenshot, 18 (0x12)-Intelligent detection and capture, 19 (0x13)-Enter the area</p> <p>Detection snapshot, 20 (0x14) - Leaving area detection snapshot, 21 (0x15) - Wandering detection capture, 22 (0x16) - Personnel</p> <p>Gathering detection capture, 23 (0x17) - fast movement detection capture</p> <p>Figure, 24 (0x18) - Parking detection capture, 25 (0x19) -</p> <p>Item left behind detection capture, 26 (0x1a) - Item taken detection</p> <p>Capture, 27 (0x1b) - License plate detection capture, 28 (0x1c)-Client uploads picture</p>
struStartTime	<u>NET_EHOME_TIME</u> start time.	
struStopTime	<u>NET_EHOME_TIME</u> end time.	
dwstartIndex	DWORD	The query start position starts from 0.
dwMaxFileCountPer DWORD		<p>The maximum number of files that can be queried in a single search is determined by the actual network environment.</p> <p>The recommended maximum number of files is 8.</p>
byLocalOrUTC	BYTE	Time type: 0-device local time, that is, device OSD time time; 1-UTC time.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 63 bytes.

A.1.100 NET_EHOME_PIXEL_ARRAY_SIZE

Heatmap size information structure

member	type of data	describe
dwLineValue	DWORD	Pixel row value.
dwColumnValue	DWORD	Pixel column value.

A.1.101 NET_EHOME_PLANCTRLPARAM

Timing control plan parameter structure

member	type of data	describe
enumPlanType	<u>NET_EHOME_PLANCTR</u> <u>L_TYPE</u>	Control type.
bEnable	BOOL	Whether to enable the control plan.

See also

[NET_EHOME_TERMINAL_CONTROL_V20](#)

A.1.102 NET_EHOME_PLAYBACK_DATA_CB_INFO

The callback information structure of playback data

member	type of data	describe
dwType	BYTE	Data type: 0-code stream header, 1-code stream data.
pData	void*	The buffer required to save the code stream header or code stream data.
dwDataLen	DWORD	The size of the code stream header or code stream data buffer.
byRes	BYTE[]	Reserved. The maximum length is 128 bytes.

A.1.103 NET_EHOME_PLAYBACK_DATA_CB_PARAM

The callback parameter structure of playback data

member	type of data	describe
fnPlayBackDataCB	<i>PLAYBACK_DATA_CB</i> callback	function for playback data.
pUserData	void*	User parameters.
byStreamFormat	BYTE	Packaging format: 0-PS.
byRes	BYTE	Reserved, set to 0.

A.1.104 NET_EHOME_PLAYBACK_INFO_IN

Input parameter structure of playback request

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwChannel	DWORD	Playback channel number.
byPlayBackMode	BYTE	Playback and download mode: 0-by file name, 1-by time.
byStreamPackage	BYTE	Stream playback type: 0-PS (default), 1-RTP.
byLinkMode	BYTE	Protocol type: 0-TCP, 1-UDP (UDP reserved), 8-NPQ
byRes	BYTE	reserve.
unionPlayBackMode Union (<i>surface 16-9</i>)	Playback download mode parameter union.
struStreamSever	<i>NET_EHOME_IPADDRESS</i>	The address information of the streaming media server.

ISUPSDK (General) Development Guide

Table A-9 Union Playback Mode (unionPlayBackMode)

member	type of data	describe
byLen	BYTE	The maximum length of the union is 512 bytes.
struPlayBackByName Struct (<u>surface 16-10</u>) Parameter structure for playback by file name.	
struPlayBackByTime Struct (<u>surface 16-11</u>) Parameter structure for playback by time.	

Table A-10 Parameter structure for playback by file name

member	type of data	describe
szFileName	char	The file name to be played back. Maximum length The length is 100 bytes (corresponding to the macro definition is MAX_FILE_NAME_LEN).
dwSeekType	DWORD	Offset calculation method: 0-byte length Degrees, 1-by time (seconds).
wF	DWORD	File offset, start playback or download s position.
dwFileSpan	DWORD	The size of the downloaded file is 0. to download the entire file.

Table A-11 Parameter structure for playback by time

member	type of data	describe
struStartTime	<u>NET_EHOME_TIME</u>	The time to start playback.
struStopTime	<u>NET_EHOME_TIME</u>	The time to stop playback.
byLocalOrUTC	BYTE	Time type: 0-device local time, That is, the device OSD time; 1-UTC time between.
byDuplicateSegment	BYTE	The position of the repeating time period: 0-repeat time The first part of the interval, 1-repeat time period

member	type of data	describe
		The latter part. When byLocalOrUTC is 1 , this parameter is invalid.

Remark

Currently only playback by file name is supported.

A.1.105 NET_EHOME_PLAYBACK_INFO_OUT

Output parameter structure of playback request

member	type of data	describe
iSessionID	LONG	The playback request session ID returned by the device: 0 - invalid.
lHandle	LONG	Message handle for asynchronous callback.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 124 bytes.

A.1.106 NET_EHOME_PLAYBACK_LISTEN_PARAM

Playback monitoring parameter structure

member	type of data	describe
strIPAddress	<u>NET_EHOME_IPADDRE</u> SS	Local monitoring information. When the IP is 0.0.0.0, the default is the local address; when there are multiple network cards, the default is the first address obtained by the operating system.
fnNewLinkCB	<u>PLAYBACK_NEWLINK_C</u> B	The callback function for replaying requests.
pUserData	void*	User parameters.

ISUPSDK (General) Development Guide

member	type of data	describe
byLinkMode	BYTE	The access protocol type of the listening request: 0-TCP, 1-UDP (protected Stay), 8-NPQ.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 127 bytes.

Remark

If the device does not support ISUP version 4.0 or above, the **SessionID** is invalid (set to 0). Please use a different port and echo.

The call function starts monitoring to distinguish the connected devices and channels.

A.1.107 NET_EHOME_PLAYBACK_NEWSLINK_CB_INFO

Playback request callback information structure (applicable to **64- bit Windows** or **Linux** operating systems)

member	type of data	describe
szDeviceID	char[]	Device ID. The maximum length is 256 bytes (corresponding to the macro definition is "NET_EHOME_DEVICEID_LEN").
lSessionID	LONG	The device session ID for the playback request, represented by <u>NET_ECMS_StartPlayBack</u> returns. If invalid, set is 0.
dwChannelNo	DWORD	Device channel number, 0-invalid.
sDeviceSerial	char[]	Device serial number. Maximum length is 12 bytes (corresponding to macro definition is defined as "NET_EHOME_SERIAL_LEN").
byStreamFormat	BYTE	Stream encapsulation format: 0-PS, 1-RTP.
byRes1	BYTE[]	Reserved. Maximum length is 3 bytes.
fnPlayBackDataCB	<u>PLAYBACK_DATA_CB</u> data callback function, its size is 8 bytes.	
pUserData	void*	User parameters, its size is 8 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 88 bytes.

Playback request callback information structure (applicable to **32-bit Windows or Linux** operating systems)

member	type of data	describe
szDeviceID	char[]	Device ID. The maximum length is 256 bytes (corresponding to the macro definition is "NET_EHOME_DEVICEID_LEN").
lSessionID	LONG	The device session ID for the playback request, represented by <u>NET_ECMS_StartPlayBack</u> returns. If invalid, set is 0.
dwChannelNo	DWORD	Device channel number, 0-invalid.
sDeviceSerial	char[]	Device serial number. Maximum length is 12 bytes (corresponding to macro definition is defined as "NET_EHOME_SERIAL_LEN").
byStreamFormat	BYTE	Stream encapsulation format: 0-PS, 1-RTP.
byRes1	BYTE[]	Reserved. Maximum length is 3 bytes.
fnPlayBackDataCB	<u>PLAYBACK_DATA_CB</u> data callback function, its size is 4 bytes.	
byRes2	BYTE[]	Reserved, the maximum length is 4 bytes.
pUserData	void*	User parameters, its size is 4 bytes.
byRes3	BYTE[]	Reserved, the maximum length is 4 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 88 bytes.

A.1.108 NET_EHOME_PLAYBACK_PAUSE_RESTART_PARAM

Playback pause or resume parameter structure

member	type of data	describe
ISessionID	LONG	The session ID for playback.
IHandle	LONG	Output parameter pointer, used as asynchronous callback in asynchronous mode Identifier, corresponding to <u>NET_EHOME_CMSCB_DATA</u> dwHandle
byRes	BYTE[]	Reserved. The maximum length is 120 bytes.

A.1.109 NET_EHOME_PLAYER_PARAM

Terminal playback parameter structure

member	type of data	describe
szCfgType	BYTE[]	Configuration type: "volume", "light" "logo"LOGO"defaultSchedule" "configuration plan"), "password", "temperature" (temperature), "allParam" (all parameters). The maximum length is 32 bytes.
dwplayerVolume	DWORD	Playback volume.
struDisplayParam	<u>NET_EHOME_DISPLAY</u> <u>PARAM</u>	Display parameters.
struPasswordCfg	<u>NET_EHOME_PASSWO</u> <u>RD_CFG</u>	Terminal password.

A.1.110 NET_EHOME_POINT

Coordinate parameter structure

member	type of data	describe
dwX	DWORD	X-axis coordinate information.
dwY	DWORD	Y-axis coordinate information.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 4 bytes.

A.1.111 NET_EHOME_POSITION_INFO

Location information structure

member	type of data	describe
wnX DWORD		The X-coordinate of the position.
wnY DWORD		The Y coordinate of the position.
dwHeight	DWORD	high.
dwWidth	DWORD	width.

A.1.112 NET_EHOME_POST_PUBLISH_XML

XML file transmission parameter structure, used to configure schedule, insert, upgrade and other parameters

member	type of data	describe
szServerIP	BYTE[]	Server IP address. The maximum length is 32 bytes.
szServerPort	DWORD	Server port number.
dwTerminalID	DWORD	Terminal ID, returned with the process.
dwXmlUniqueSeq	DWORD	The XML file ID.
dwXmlSizeH	DWORD	XML file size, represented by the high 32 bits.

member	type of data	describe
dwXmlSizeL	DWORD	The size of the XML file is represented by the lower 32 bits.
byStorageId	BYTE[]	The save path of the XML file. The maximum length is 128 characters. Festival.
byRes	BYTE[]	Reserved. The maximum length is 16 bytes.

A.1.113 NET_EHOME_POST_SCHEDULE

Schedule delivery parameter structure

member	type of data	describe
byServerIP	BYTE[]	Server IP address. The maximum length is 32 bytes.
dwServerPort	DWORD	Server port number.
dwScheduleID	DWORD	The newly created schedule is the assigned schedule ID.
dwScheduleSeq	DWORD	Indicates whether the schedule has been modified and whether the transmission can continue. Effective for ANR.
byIsDefaultSchedule BOOL		Indicates whether it is a shim schedule.
byRes	BYTE[]	Reserved. The maximum length is 2 bytes.
dwTimingScheduleID DWORD		The schedule ID.
byCVM BYTE[]		The time when the schedule starts playing. The maximum length is 32 bytes.

A.1.114 NET_EHOME_PPPOECFG

PPPoE parameter structure

member	type of data	describe
dwPPPoE	DWORD	Whether to enable PPPoE: 1-Yes, 0-No.
sPPPoEUser	char[]	PPPoE user name. The maximum length is 32 bytes (corresponding to macro is defined as "NET_EHOME_SERIAL_LEN").
sPPPoEPASSWORD	char[]	PPPoE password. The maximum length is 16 bytes (corresponding to the macro definition is defined as "PASSWD_LEN").
struPPPoEIP	<u>NET_EHOME_IPADDR</u>	The IP address of PPPoE.

A.1.115 NET_EHOME_PRESET_PARAM

Preset point parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byPresetCmd	BYTE	Preset point control commands: 1-set preset point, 2-delete preset point, 3-call preset point.
byRes1	BYTE[]	Reserved, set to 0. The maximum length is 3 bytes.
dwPresetIndex	DWORD	Preset point number.
byRes2	BYTE[]	Reserved, set to 0. The maximum length is 32 bytes.

A.1.116 NET_EHOME_PREVIEW_CB_MSG

Preview callback data structure

member	type of data	describe
byDataType	BYTE	Data type: 1-NET_DVR_SYSHEAD (stream header), 2-NET_DVR_STREAMDATA (stream data).
byRes1	BYTE[]	Reserved. The maximum length is 3 bytes.
pRecvdata	void*	A buffer that stores the code stream header or code stream data.
dwDataLen	DWORD	The size of the code stream header or code stream data buffer.
byRes2	BYTE[]	Reserved. The maximum length is 128 bytes.

A.1.117 NET_EHOME_PREVIEW_DATA_CB_PARAM

Preview callback parameter structure

member	type of data	describe
fnPreviewDataCB	<i>PREVIEW_DATA_CB</i> callback function for preview.	
pUserData	void*	User parameters.
byStreamFormat	BYTE	Stream encapsulation format: 0-PS
byRes	BYTE[]	Reserved, set to 0. The maximum length is 127 bytes.

A.1.118 NET_EHOME_PREVIEWINFO_IN

Preview request input parameter structure

member	type of data	describe
iChannel	int	Channel number.
dwStreamType	DWORD	Stream type: 0-main stream, 1-sub stream, 2-tertiary stream.

ISUPSDK (General) Development Guide

member	type of data	describe
dwLinkMode	DWORD	<p>Access mode: 0-TCP, 1-UDP, 2-HRUDP (reliable transmission lose), 8-NPQ</p> <p> illustrate</p> <p>The use of NPQ function is currently combined with Smart and variable bit rate. If you want to use the NPQ function, you must set The bit rate is adjusted to a fixed bit rate, the Smart function is turned off, 8111 The error code serves as an error prompt for modifying the configuration.</p>
struStreamSever	<u>NET_EHOME_IPADDRE</u> SS	The address information of the streaming media server.

A.1.119 NET_EHOME_PREVIEWINFO_IN_V11

Preview request input parameter structure

member	type of data	describe
iChannel	int	Channel number.
dwStreamType	DWORD	Stream type: 0-main stream, 1-sub stream, 2-tertiary stream.
dwLinkMode	DWORD	Access mode: 0-TCP, 1-UDP, 2-HRUDP (reliable transmission lose)
struStreamSever	<u>NET_EHOME_IPADDRE</u> SS	The address information of the streaming media server.
byDelayPreview	BYTE	Preview mode: 0-get real-time stream, 1-get delayed stream flow.
byEncrypt	BYTE	Whether the code stream is encrypted: 0-No, 1-Yes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 30 bytes.

A.1.120 NET_EHOME_PREVIEWINFO_OUT

ISUPSDK (General) Development Guide

Output parameter structure of preview request

member	type of data	describe
lSessionID	int	The session ID for the preview request.
lHandle	DWORD	Message handle for asynchronous stream fetch callback.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 124 bytes.

A.1.121 NET_EHOME_PT_PARAM

Listening agent parameter structure

member	type of data	describe
strIP	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	Listening address information (IP address and port number).
byProtocolType	BYTE	Protocol type: 0-TCP
byProxyType	BYTE	Proxy type: 0-Hikvision private protocol proxy, 1-HTTP proxy reason.
byRes	BYTE[]	Reserved. The maximum length is 2 bytes.

A.1.122 NET_EHOME_PTXML_PARAM

Transmission parameter structure

member	type of data	describe
pRequestUrl	void*	Request URL.
dwRequestUrlLen	DWORD	Request URL size.
CnB	void*	A buffer that holds conditional parameters (in XML or JSON format) district.

ISUPSDK (General) Development Guide

member	type of data	describe
dwCondSize	DWORD	Condition buffer size.
nB	void*	A buffer to store input parameters (in XML or JSON format) district.
dwInSize	DWORD	Enter the buffer size.
B	void*	Buffer to store output parameters (in XML or JSON format) district.
dwOutSize	DWORD	Output buffer size. The space allocated by the output buffer is full. When it is sufficient, it indicates the size of the returned message data.
dwReturnedXMLLen DWORD		The actual size of the received XML or JSON data. Output Buffer If the space in the area is insufficient, it means that the returned message data is too large. Small.
dwRecvTimeOut	DWORD	Receive timeout.
dwHandle	DWORD	Message handle. After setting the playback asynchronous callback, Used for identification.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 24 bytes.

A.1.123 NET_EHOME_PTZ_PARAM

PTZ parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
byPTZCmd	BYTE	PTZ control commands, see notes.
bycn	BYTE	PTZ control: 0-start, 1-stop.

ISUPSDK (General) Development Guide

member	type of data	describe
bySpeed	BYTE	PTZ speed, the value range is from 0 to 70. The larger the value, the faster the PTZ speed. The faster the speed.
byRes	BYTE[]	Reserved. The maximum length is 29 bytes.

PTZ control commands

dwCommand	Command number	describe
PTZ_UP	0	up
PTZ_DOWN	1	down
PTZ_LEFT	2	left
PTZ_RIGHT	3	To the right
PTZ_UPLEFT	4	Up left
PTZ_DOWNLEFT	5	Down left
PTZ_UPRIGHT	6	Up right
PTZ_DOWNRIGHT	7	Down right
PTZ_ZOOMIN	8	Zoom (zoom out)
PTZ_ZOOMOUT	9	Zoom (enlarge)
PTZ_FOCUSNEAR	10	Focus -
PTZ_FOCUSFAR	11	Focus+
PTZ_IRISSTARTUP	12	Aperture becomes larger
PTZ_IRISSTOPDOWN	13	Smaller aperture
PTZ_LIGHT	14	Fill Light
PTZ_WIPER	15	Wipers
PTZ_AUTO	16	automatic

A.1.124 NET_EHOME_PUBLISH_SERVERADDR

Information publishing server address information structure

member	type of data	describe
enumAddressingForm	<u>NET_EHOME_ADDRFO</u>	The address type.
atType	<u>RMAT_CMD</u>	
szIpAddress	BYTE[]	IP address. The maximum length is 32 bytes (the corresponding macro is defined as "NET_EHOME_MAX_TYPE_LEN").
szIpv6Address	BYTE[]	IPv6 address. The maximum length is 32 bytes (corresponding to the macro definition is "NET_EHOME_MAX_TYPE_LEN").
szHostName	BYTE[]	Domain name. The maximum length is 32 bytes (the corresponding macro is defined as "NET_EHOME_MAX_TYPE_LEN").
dwPortNo	DWORD	The port number.
szUserName	BYTE[]	Username. Maximum length is 32 bytes (the corresponding macro is defined as "NET_EHOME_MAX_TYPE_LEN").
szPassword	BYTE[]	Password. The maximum length is 32 bytes (the corresponding macro is defined as "NET_EHOME_MAX_TYPE_LEN").
bRegStatus	BOOL	Registration status.

A.1.125 NET_EHOME_PUSHPLAYBACK_IN

Input parameter structure of playback stream transmission request

member	type of data	describe
dwSize	DWORD	The size of the structure.
lSessionID	LONG	Session ID, returned by <u>NET_ECMS_StartPlayBack</u> .

ISUPSDK (General) Development Guide

member	type of data	describe
byKeyMD5	BYTE[]	The code stream encryption key is obtained by two MD5 calculations. The maximum length is 32 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 96 bytes.

A.1.126 NET_EHOME_PUSHPLAYBACK_OUT

Output parameter structure of playback stream transmission request

member	type of data	describe
dwSize	DWORD	The size of the structure.
lHandle	LONG	Message handle for asynchronous callback.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 124 bytes.

A.1.127 NET_EHOME_PUSHSTREAM_IN

Input parameter structure for real-time streaming request

member	type of data	describe
dwSize	DWORD	The size of the structure.
lSessionID	LONG	The session ID for the preview request, represented by <u>NET_ECMS_StartGetRealStreamV11</u> returns.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 128 bytes.

A.1.128 NET_EHOME_PUSHSTREAM_OUT

ISUPSDK (General) Development Guide

Output parameter structure of real-time streaming request

member	type of data	describe
dwSize	DWORD	The size of the structure.
IHandle	LONG	Message handle for asynchronous stream fetch callback.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 124 bytes.

A.1.129 NET_EHOME_PUSHVOICE_IN

Input parameter structure of audio code stream transmission request

member	type of data	describe
dwSize	DWORD	The size of the structure.
ISessionID	LONG	The session ID of the voice intercom request, represented by <u>NET_ECMS_StartVoiceWithStmServer</u> returns.
byToken	BYTE[]	Token information. The maximum length is 64 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 64 bytes.

A.1.130 NET_EHOME_PUSHVOICE_OUT

Output parameter structure of audio code stream transmission request

member	type of data	describe
dwSize	DWORD	The size of the structure.
IHandle	LONG	Message handle for asynchronous audio stream callback.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 124 bytes.

A.1.131 NET_EHOME_PZIN_PARAM

Zoom parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
bycn	BYTE	Zoom type: 0-zoom out, 1-zoom in.
byRes1	BYTE[]	Reserved, set to 0. The maximum length is 3 bytes.
struArea	<u>NET_EHOME_ZONE</u> box selection	range parameters.
byRes2	BYTE[]	Reserved, set to 0. The maximum length is 32 bytes.

A.1.132 NET_EHOME_QUEUE_DATA

Calling number data structure

member	type of data	describe
szQueueDataValue	char[]	Calling number. The maximum length is 32 bytes.
dwQueueDataId	DWORD	Caller ID.

A.1.133 NET_EHOME_QUEUE_DATALIST

Call number list structure

member	type of data	describe
queueData	<u>NET_EHOME_QUEUE_</u> <u>DATA []</u>	Calling number. Maximum length is 20 bytes.

A.1.134 NET_EHOME_QUEUE_INFO

Call management information structure

member	type of data	describe
dwMaterialId	DWORD	The material ID.
dwQueueId	DWORD	Caller ID.
enumDataType	<u>NET_EHOME_THIRD_P</u> <u>ARTY_DATA_TYPE</u>	Command type.
enumRefreshType	<u>NET_EHOME_REFRESH</u> <u>TYPE</u>	Data refresh type.
strulItemDataList	<u>NET_EHOME_QUEUE_I</u> <u>TEM_DATALIST</u>	Call number list.

A.1.135 NET_EHOME_QUEUE_ITEM_DATALIST

Call number list structure

member	type of data	describe
dwQueueDataListCnt DWORD		The number of rows in the list.
dwQueueDataCnt	DWORD	The number of columns in the list.
struQueueDatalist	<u>NET_EHOME_QUEUE_</u> <u>DATALIST[]</u>	Call number list. Maximum length is 4 bytes.

A.1.136 NET_EHOME_REC_FILE

Video file search result structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
szFileName	char[]	Video file name. Maximum length is 100 bytes (corresponding to macro definition is defined as "MAX_FILE_NAME_LEN").
struStartTime	<u>NET_EHOME_TIME</u> start time.	
struStopTime	<u>NET_EHOME_TIME</u> end time.	
dwFileSize	DWORD	Video file size.
dwFileMainType	DWORD	Recording type: 0x all recording types, 0- continuous recording, 1-Motion detection recording, 2-Alarm trigger recording, 3-Alarm trigger 4- alarm and motion detection recording, 5-command triggered recording, 6-manual recording, 7-vibration alarm recording Image, 8-environmental alarm recording, 9-intelligent alarm recording (or 10 (0x0a)-PIR alarm recording, 11 (0x0b)-wireless alarm recording, 12 (0x0c)-call for help Video recording, 13 (0x0d) - all types of alarm recordings.
dwFileSubType	DWORD	Subtype of video file: 0-non-traffic evidence video, 1-super Car, 2-occupying the road, 3-illegal parking, 4-reversing, 5-reversing, 6-U-turn, 7-Lane change, 8-Crossing the line, 9-Others; only when This parameter is valid when dwFileMainType is set to 1.
dwFileIndex	DWORD	Video file number, starting from 0.
l would	BYTE	The difference between the start/end time and UTC time (hours), value It is -12, -10, ..., +12, +14, and 0x indicates invalid.
bymM BYTE		The difference between the start/end time and UTC time (minutes), value -30, 0, 30, 45, 0x indicates invalid.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 126 bytes.

A.1.137 NET_EHOME_REC_FILE_COND

Video file search condition structure

member	type of data	describe
dwChannel	DWORD	Channel number.
dwRecType	DWORD	Recording type: 0x all recording types, 0- continuous recording, 1-Motion detection recording, 2-Alarm trigger recording, 3-Alarm trigger 4- alarm and motion detection recording, 5-command triggered recording, 6-manual recording, 7-vibration alarm recording Image, 8-environmental alarm recording, 9-intelligent alarm recording (or 10 (0x0a)-PIR alarm recording, 11 (0x0b)-wireless alarm recording, 12 (0x0c)-call for help Video recording, 13 (0x0d) - all types of alarm recordings.
struStartTime	<u>NET_EHOME_TIME</u> start time.	
struStopTime	<u>NET_EHOME_TIME</u> end time.	
dwstartIndex	DWORD	The starting position of the query, starting from 0.
dwMaxFileCountPer DWORD		The maximum number of files that can be queried in a single search is determined by the actual network environment. The recommended maximum number of files is 8.
byLocalOrUTC	BYTE	Time type: 0-device local time, that is, device OSD time time; 1-UTC time.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 63 bytes.

A.1.138 NET_EHOME_RECORD_CHAN

ISUPSDK (General) Development Guide

Associated video channel parameter structure

member	type of data	describe
byAnalogChanNum	BYTE	Number of analog channels, read-only.
byAnalogChan	BYTE[]	Analog channel status, represented by array subscripts: 0-disabled, 1- Enabled. The maximum length is 32 bytes (the corresponding macro is defined as "MAX_ANALOG_CHANNUM")
byRes1	BYTE[]	Reserved, set to 0. The maximum length is 3 bytes.
wDigitChanNum	WORD	Number of digital channels, read-only.
byDigitChan	BYTE[]	Digital channel status, represented by array subscript, 0-not enabled, 1 - Enabled. Maximum length is 480 bytes.
byRes2	BYTE[]	Reserved, set to 0. The maximum length is 62 bytes.

A.1.139 NET_EHOME_REGISTER_LISTEN_MODE

Listening mode structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwRegisterListenMode DWORD		Listening mode: 0-TCP and UDP, 1-UDP, 2-TCP.
byRes	BYTE[]	Reserved. The maximum value length is 128 bytes.

A.1.140 NET_EHOME_RELATE_IPC_INFO

Related network camera information structure

member	type of data	describe
dwIpcNum	DWORD	The number of associated network cameras.
strIpcInfo	<u>NET_EHOME_IPC_INF</u> [O[]]	Network camera information. The maximum length is 6 bytes.

A.1.141 NET_EHOME_RELEASE_SERVER_INFO

Information publishing system information structure

member	type of data	describe
dwKeepAliveSeconds DWORD		Keep-alive time, in seconds.
byAlarmServerIP	BYTE[]	The IP address of the alarm management server. The maximum length is 32 byte.
dwAlarmServerPort DWORD		The port number of the Alarm Management Server.
dwAlarmServerType DWORD		Communication type of the alarm management server: 0-TCP (default), 1-UDP
dwAlarmServerTcpPort	DWORD	Alarm port sent by the alarm management server to the terminal Number.
byNtpServerIP	BYTE[]	The IP address of the NTP server. The maximum length is 32 bytes.
dwNtpServerPort	DWORD	The port number of the NTP server.
dwNtpInterval	DWORD	NTP time synchronization interval.
byPicServerIP	BYTE[]	The IP address of the storage server. The maximum length is 32 bytes.
dwPicServerPort	DWORD	The port number of the storage server.
dwPicServerType	DWORD	Communication type of the storage server: 0-TCP (default), 1-UDP

ISUPSDK (General) Development Guide

member	type of data	describe
byBlackListAddr	BYTE[]	The address of the unauthorized list management server. The maximum length is 32 bytes.
byBlackListName	BYTE[]	The name of the unauthorized list management server. The maximum length is 32 bytes.
dwBlackListPort	DWORD	The port number of the unauthorized list management server.
byBlackListUser	BYTE[]	User name of the unauthorized list management server. Maximum length is 32 bytes.
byBlackListPasswd	BYTE[]	The password for the unauthorized list management server. The maximum length is 32 bytes.
dwTranserSerialSvrPort DWORD		The port number of the transport server.
bReliableTransmission BOOL		Whether to enable reliable transmission: True-Yes, False-No.

A.1.142 NET_EHOME_REMOTE_CTRL_PARAM

Control parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
CnB	void*	The buffer for storing conditional parameters is <i>NET ECMS_RemoteControl</i> control commands (dwCommand) determines the command. See the table in the Remarks.
wCnB	DWORD	Condition parameter buffer size.
nb	void*	The buffer for saving control parameters is <i>NET ECMS_RemoteControl's (dwCommand)</i> For details, see the table in the notes.

ISUPSDK (General) Development Guide

member	type of data	describe
wnB	DWORD	Controls the parameter buffer size.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 32 bytes.

Remark

The relationship among **dwCommand**, **CnB** and **nb** is shown in the following table.

dwCommand	CnB	nb
NET_EHOME_MANUAL_IOOUT	Null	<u>NET_EHOME_MANUAL_IOOUT_CTRL</u>
NET_EHOME_PTZ_CTRL	4-byte channel number ÿlongÿ	<u>NET_EHOME_PTZ_PARAM</u>
NET_EHOME_PRESET_CTRL	4-byte channel number ÿlongÿ	<u>NET_EHOME_PRESET_PARAM</u>
NET_EHOME_PZIN	4-byte channel number ÿlongÿ	<u>NET_EHOME_PZIN_PARAM</u>
NET_EHOME_PTRACK	4-byte channel number ÿlongÿ	<u>NET_EHOME_POINT</u>

A.1.143 **NET_EHOME_REPLACE_MATERIAL**

Material replacement parameter structure

member	type of data	describe
szServerIP	BYTE[]	Server IP address. Maximum length is 32 bytes (corresponding to macro Defined as "NET_EHOME_MAX_TYPE_LEN").
dwServerPort	DWORD	Server port number.
dwMaterialId	DWORD	The material ID.
dwMaterialSeq	DWORD	Material number.

A.1.144 NET_EHOME_SCHEDTIME

Time period parameter structure

member	type of data	describe
byStartHour	BYTE	Start time: .
byStartMin	BYTE	Start time: minutes.
byStopHour	BYTE	End time: hours.
byStopMin	BYTE	End time: minutes.

A.1.145 NET_EHOME_SCREEN_SHOT

Terminal screenshot parameter structure

member	type of data	describe
dwServerPort	DWORD	Server port number.
dwClientFdIndex	DWORD	Client FD index number.

A.1.146 NET_EHOME_SCREEN_SHOT_EX

Terminal screenshot parameter structure

member	type of data	describe
dwServerPort	DWORD	Server port number.
dwClientFdIndex	DWORD	Client FD index number.
szKmsURL	BYTE[]	The URL address of the screenshot on the KMS server. The length is 256 bytes.

A.1.147 NET_EHOME_SCREEN_SHOT_RET

Terminal screenshot response structure

member	type of data	describe
dwConnfd	unsigned int	Socket for asynchronous communication between terminal and server.
dwSeq	unsigned int	ID for asynchronous communication between the terminal and the server.
sUUID	char[]	The UUID of the screenshot image. The maximum length is 64 bytes.
szPicURL	char[]	The URL address of the screenshot image on the KMS server. The length is 256 bytes.

A.1.148 NET_EHOME_SEND_PARAM

Data sending parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwRecvTimeOut	DWORD	Receive timeout, unit: ms. The value range is 0 or [5000,60000]. When it is 0, it means using the default receiving Timeout period (5000 ms).
bySendTimes	BYTE	The number of times a message is sent repeatedly can reduce packet loss when the network is poor. The value range is 1, 2, or 3. The default value is 1. Maximum 3 times.
byRes2	BYTE[]	Reserved. The maximum length is 127 bytes.

A.1.149 NET_EHOME_SERVER_INFO

Server information structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwKeepAliveSec	DWORD	Heartbeat interval, unit: seconds, default: 15 seconds.
dwTimeOutCount	DWORD	The default value is 0, which means the heartbeat timeout is The number of times is 6 times.
struTCPAlarmSever <i>NET_EHOME</i>	<i>IPADDRE</i> SS	The TCP address of the Alarm Management Server.
struUDPAAlarmSever <i>NET_EHOME</i>	<i>IPADDRE</i> SS	UDP address of the Alarm Management Server.
dwAlarmServerType DWORD		Alarm Management Server (AMS) Type: 0-supports only UDP, 1-supports UDP and TCP, 2-supports MQTT (server Server address is based on TCP).
struNTPSever	<i>NET_EHOME_IPADDRE</i> SS	NTP server address information.
dwNTPInterval	DWORD	NTP time synchronization interval, in seconds.
struPictureSever	<i>NET_EHOME_IPADDRE</i> SS	Storage server address information.
dwPicServerType	DWORD	Storage Server (SS) Type: 0-Tomcat, 1-VRB, 2- Cloud storage, 3-KMS, 4-ISUP5.0.
struBlackListServer	<i>NET_EHOME_BLACKLIS</i> <i>T_SEVER</i>	Unauthorized list management server.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 128 bytes.

A.1.150 NET_EHOME_SERVER_INFO_V50

Server information structure (V50)

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwKeepAliveSec	DWORD	Heartbeat interval, unit: seconds, default: 15s.
dwTimeOutCount	DWORD	The default value is 0, which means the heartbeat timeout is The number of times is 6 times.
struTCPAlarmSever <i>NET_EHOME_IPADDRE</i> SS		The TCP address of the Alarm Management Server.
struUDPAlarmSever <i>NET_EHOME_IPADDRE</i> SS		UDP address of the Alarm Management Server.
dwAlarmServerType DWORD		Alarm Management Server (AMS) Type: 0-supports only UDP, 1-supports UDP and TCP, 2-supports MQTT (server Server address is based on TCP).
struNTPSever	<i>NET_EHOME_IPADDRE</i> SS	NTP server address information.
dwNTPInterval	DWORD	NTP time synchronization interval, in seconds.
struPictureSever	<i>NET_EHOME_IPADDRE</i> SS	Storage server address information.
dwPicServerType	DWORD	Storage Server (SS) Type: 0-Tomcat, 1-VRB, 2- Cloud storage, 3-KMS, 4-ISUP5.0.
struBlackListServer	<i>NET_EHOME_BLACKLIS</i> <i>T_SEVER</i>	Parameters of the unauthorized list management server.
struRedirectSever	<i>NET_EHOME_IPADDRE</i> SS	The address of the redirect server.
byCloudAccessKey	BYTE[]	The access password of the cloud storage. The maximum length is 64 bytes.
byCloudSecretKey	BYTE[]	The key for cloud storage. The maximum length is 64 bytes.

member	type of data	describe
to be	BYTE	Enable cloud storage type: 1-HTTPS, 0-HTTP.
byRes1	BYTE[]	Reserved. The maximum length is 3 bytes.
dwAlarmKeepAliveSec DWORD		Alarm heartbeat interval, unit: seconds, default: 30s.
dwAlarmTimeOutCount	DWORD	The default setting is 0, which means the alarm heartbeat timeout is The number of jump timeouts is 3 times.
dwCloudPoolId	DWORD	Cloud storage pool ID.
byRes	BYTE[]	Reserved. The maximum length is 368 bytes.

A.1.151 NET_EHOME_SET_REREGISTER_MODE

Device registration mode structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwReRegisterMode DWORD		Whether to support repeated registration (only supports ISUP version 2.0): 0-Allow 2.0 protocol devices to repeat during the keep alive time Registration (default), 1-Do not allow 2.0 protocol during keepalive time If the device is registered repeatedly, the error code "403" will be returned.

A.1.152 NET_EHOME_SHOT_PIC

Pop-up command parameter structure

member	type of data	describe
byServerIP	BYTE[]	Server IP address. The maximum length is 32 bytes.
dwServerPort	DWORD	Server port number.
dwShotPicID	DWORD	Popup image ID.

A.1.153 NET_EHOME_SS_CENTRAL_PARAM

Central storage complex

member	type of data	describe
pPoolid	const char*	Resource pool ID, with a maximum length of 32 bytes.
pSerialID	const char*	Device ID, the maximum length is 64 bytes.
pSerialIDUUID	const char*	SerialID+UUID represents the key of the video file, maximum length 64 bytes (device ID length) + 32 bytes (UUID length Every time).
pBeginTime	const char*	Starting time.
pEndTime	const char*	End Time.
byPoolidLength	BYTE	Resource pool ID length, only valid when uploading files.
bySerialIDLength	BYTE	Device ID length, only valid when uploading files.
bySerialIDUUIDLength BYTE		Video file key length, only when the file type is video file efficient.
byBeginTimeLength	BYTE	Start time length.
byEndTimeLength	BYTE	End time length.
byTransform	BYTE	Whether to need to repackage: 0-not needed (default), 1-needed.
byRes1	BYTE[]	Reserved, the maximum length is 2 bytes.

member	type of data	describe
dwRecordType	int	Video recording type.
dwSourceDataType	int	Video file type: 4-Hikvision video file (stored internally) Line frame analysis), other values - third-party video files.
dwHeadSize	int	Video head length.
dwErrorCode	int	The center stores the error code.
byRes	BYTE[]	Reserved. The maximum length is 468 bytes.

A.1.154 NET_EHOME_SS_CLIENT_PARAM

File upload client parameter structure

member	type of data	describe
enumType	<u>NET_EHOME_SS_CLIE</u> <u>N_TYPE</u>	File upload client type.
struAddress	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	The IP address of the storage server (SS).
by	BYTE	Whether to enable HTTPS communication.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 63 bytes.

Related interfaces

[NET_ESS_CreateClient](#)

A.1.155 NET_EHOME_SS_CLOUD_PARAM

Cloud storage structure

member	type of data	describe
pPoolId	const char*	Resource pool ID.
byPoolIdLength	BYTE	Resource pool ID length.
dwErrorCode	int	Cloud storage error code.
byRes	BYTE[]	Reserved. Maximum length is 503 bytes.

A.1.156 NET_EHOME_SS_EX_PARAM

The extended structure returned by the callback function

member	type of data	describe
byProtoType	BYTE	Storage protocol type 1- cloud storage, 2- TOMCAT, 3- KMS, 4-VRB
byRes	BYTE[]	Reserved. Maximum length is 23 bytes.
<u>surface 16-12</u>	union	Consortium.

ÿ A-12 unionStoreInfo

member	describe
struCentral	NET_EHOME_SS_CENTRAL_PARAM
struCloud	NET_EHOME_SS_CLOUD_PARAM
struTomcat	NET_EHOME_SS_TOMCAT_PARAM
struKms	NET_EHOME_SS_KMS_PARAM
struVrb	NET_EHOME_SS_VRB_PARAM

A.1.157 NET_EHOME_SS_INIT_PARAM

Initialization parameter structure

member	type of data	describe
szKey	char[]	Database key (string type), maximum length is 32 characters Section (the corresponding macro is defined as "SS_DB_KEY_MAX_LEN").
byRes	BYTE[]	Reserved. Maximum length is 224 bytes.

Remark

- Allow NULL parameters to be passed when calling the NET_ESS_Init_V11 interface, which is equivalent to calling the NET_ESS_Init interface.
Do not set the database key.
- The database generated by HCISUPSS is PicttrueStorageServer.db. If the database already exists during the runtime,
If encryption is performed (the key has been set), the NET_ESS_Init_V11 interface must be called and the correct key must be passed.
Can complete the initialization of HCISUPSS.
- When running, if the database already exists, if the database is not encrypted (no key is set), you cannot call
NET_ESS_Init_V11 sets the key for an existing non-encrypted database, which can only be opened in a non-encrypted manner.
- Allow storage components to call NET_ESS_GetLastError to get the error code after initialization fails, such as calling
If the correct password is not set in NET_ESS_Init_V11, the error code obtained by calling NET_ESS_GetLastError is 1.
(NET_DVR_PASSWORD_ERROR) (Password error). When the database has been set with a key, but the
NET_ESS_Init passes an empty key, so the error code is still 1 (NET_DVR_PASSWORD_ERROR) (password error
error).
- szKey** is not allowed to be empty, otherwise the NET_ESS_Init_V11 call will fail with the error code parameter error (error code 17).

A.1.158 NET_EHOME_SS_KMS_PARAM

KMS storage information structure

member	type of data	describe
byRes	BYTE[]	Reserved. Maximum length is 512 bytes.

A.1.159 NET_EHOME_SS_LISTEN_HTTPS_PARAM

ISUPSDK (General) Development Guide

HTTPS listening parameter structure

member	type of data	describe
by	BYTE	Whether to enable HTTPS: 0-disable, 1-enable.
byVerifyMode	BYTE	Verification mode: 0-single item verification.
byCcFy BYTE		Certificate type: 0-PEM, 1-ANS1
byPrivateKeyFileType BYTE		Private key type: 0-PEM, 1-ANS1
CcF char[]		Public key certificate storage path. The maximum length is 260 bytes (for should be defined as "MAX_PATH").
szUserPrivateKeyFile char[]		The private key certificate storage path. The maximum length is 260 bytes (for should be defined as "MAX_PATH").
dwSSLVersion	DWORD	SSL mode: 0-SSL23, 1-SSL2, 2-SSL3, 3-TLS1.0, 4-TLS1.1, 5-TLS1.2. SSL2 and SSL3 modes are compatible, Compatible with the latest versions of clients and servers.
byRes3	BYTE[]	Reserved. The maximum length is 360 bytes.

A.1.160 NET_EHOME_SS_LISTEN_PARAM

Listening parameter structure

member	type of data	describe
struAddress	<u>NET_EHOME_IPADDRE</u> SS	Local monitoring information. When the IP is 0.0.0.0, the default is the local address; when there are multiple network cards, the default is The first address obtained by the operating system.
szKMS_UserName	char[]	The maximum length is 512 bytes (corresponding to macro defined as "MAX_KMS_USER_LEN").
szKMS_Password	char[]	The maximum length of the KMS password is 512 bytes (corresponding to the macro is defined as "MAX_KMS_USER_LEN").

ISUPSDK (General) Development Guide

member	type of data	describe
fnSSStorageCb	<u>EHomeSSMsgCallBack</u> is the callback function for storing information.	
fnSMsgCb	<u>EHomeSSMsgCallBack</u> callback function for storing files.	
szAccessKey	char[]	AccessKey of ISUP5.0. The maximum length is 64 bytes (The corresponding macro is defined as "MAX_CLOUD_AK_SK_LEN").
szSecretKey	char[]	ISUP5.0 SecretKey. Maximum length is 64 bytes (The corresponding macro definition is "").
pUserData	void*	User parameters.
by	BYTE	Whether to enable HTTPS communication: 0-disable, 1-enable.
byRes1	BYTE[]	Reserved. The maximum length is 3 bytes.
fnSSRWCb	<u>EHomeSSRWCallBack</u> reads and writes callback functions.	
fnSSRWCbEx	<u>EHomeSSRWCallBackE</u> x —	Read and write callback function extension.
bySecurityMode	BYTE	Security mode switch, enabled by default, 0-enabled, 1-disabled.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 51 bytes.

Remark

The read-write callback function (EHomeSSRWCallBack) and the storage callback function (EHomeSSStorageCallBack) are mutually exclusive
If both callback functions are registered and configured, the default is EHomeSSRWCallBack .

Take effect.

Related interfaces

NET_ECMS_StartListen

A.1.161 NET_EHOME_SS_LOCAL_SDK_PATH

The local mount path structure of the storage server

member	type of data	describe
sPath	char[]	Load path. The maximum length is 260 bytes (corresponding to macro definition is MAX_PATH).
byRes	BYTE[]	Reserved. The maximum length is 128 bytes.

A.1.162 NET_EHOME_SS_RW_PARAM

Read and write extended callback function structure

member	type of data	describe
pFileName	const char*	file name.
pFileBuf	void*	document content.
dwFileLen	DWORD*	File size.
pFileUrl	const char*	<p>The URL of the file.</p> <p> illustrate</p> <p>The URL obtained by KMS storage method contains the token parameter The token parameter must be deleted before downloading files.</p>
navel	void*	User data.
byAct	BYTE	Read and write operations: 0-write file, 1-read file, delete file.
byUseRetIndex	BYTE	Whether to use the index set by the upper layer (pRetIndex): 0-No Use, 1-Use.
byRes1	BYTE[]	Reserved. The maximum length is 2 bytes.
pRetIndex	char*	<p>The index set by the upper layer, when the byUseRetIndex value is 0</p> <p>It can be left unset when the byUseRetIndex value is 1; it can be set when the byUseRetIndex value is 1.</p>
byRes	BYTE[]	Reserved. The maximum length is 56 bytes.

A.1.163 NET_EHOME_SS_STORAGE_URI**URI** information structure for storing resources

member	type of data	describe
enumType	<i>NET_EHOME_SS_TYPE</i> [IN]	Storage service type.
szFilename	char[]	[IN] File identifier, the maximum length is 256 bytes.
sUri	char[]	[OUT] The generated URL has a maximum length of 4096 bytes (for should be defined as "MAX_URL_LEN_SS").
byRes	BYTE[]	Reserved. Maximum length is 64 bytes.

A.1.164 NET_EHOME_SS_TOMCAT_MSG**Tomcat** server information structure

member	type of data	describe
szDevUri	char[]	Device request URL, such as: "/service/c vrcn01tim20130722201 12701qnc01n12345 cr0Wn130345N Nti28650rctin289860 y0rmy0tim20130409 090851000cnm2y0VcC r0Vcy0Cry0r1", (16-13). The maximum length is 4096 bytes (corresponding to the macro is defined as "MAX_URL_LEN_SS").
dwPicNum	DWORD	Number of images.
pPicURLs	char*	The URL of the image.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 64 bytes.

Table A-13 Field definitions in the URL

Fields	definition
trees	Device registration ID.
chanCapture	Image capture channel number.
m	Snap time
sequence	Continuously capture the number.
platinum	number plate.
platecolor	License plate color.
gpsEW	E: Eastern Hemisphere, W: Western Hemisphere.
longitude	Longitude, calculated as: degrees \times 3600 \times 100 + minutes \times 60 \times 100 + seconds \times 100.
gpsNS	N: Northern Hemisphere, S: Southern Hemisphere.
	Latitude, calculated as: degrees \times 3600 \times 100 + minutes \times 60 \times 100 + seconds \times 100.
cn	Direction, calculated as: actual direction value (north direction value is 0, unit: degree) \times 100.
speed	Speed, unit: cm/h.
datatype	Data type: 0-real-time data, 1-historical data.
alarm type	Traffic violation alarm type: 0-normal vehicle passing, 1-unauthorized List of vehicles with the right to pass.
m	Capture time format: "YYYYMMDDHHMMSSXXX" ("year month day hour minute second XXX"), the last three digits are milliseconds, such as If millisecond accuracy is not required, the last three digits can be set to "000".
picnum	Optional, number of pictures, expansion is recommended.
plateType	License plate type.
VehicleColor	Vehicle color.

Fields	definition
VehicleType	Vehicle Type.
CarType	Type of sports target.
pictype	Image type: 0-license plate image, 1-scene image, 2-synthesized picture.
piclen	Optional, image size, unit: bytes.
picname	Optional, picture name, format is "year month day hour minute second_Continuation capture number (2-digit)_PUID_channel number (2-bit).jpg".
uploadtype	Uploading mode: 0-(PU) Scheduled snapshot upload to 1- (PU) alarm capture uploaded to the center, 2- (CU) Manually capture and upload pictures to the center, 3- (CU) download pictures, 4- (PU) I/O triggers the capture to be uploaded to the center.
picid	Image ID, including time, device ID and channel number.
alarm	Alarm type: 0-hard disk full alarm, 1-hard disk error alarm Alarm, 5-Video loss alarm, 6-Alarm input alarm, 7-Video Frequency masking alarm, 8-motion detection alarm, 10-speed alarm Alarm, ..., 0xA00e-ADAS alarm, 0xA031-radar alarm police.
alarmvideochan	Alarm video channel number.
alarmdiskchan	Alarm hard disk number.
mm	Alarm time.
faceanalysis	Optional: whether to trigger image recognition capture: 0-No, 1-Yes.
c	Optional: face image file.
facerect	Optional, face frame.

A.1.165 NET_EHOME_SS_TOMCAT_PARAM

TOMCAT storage information structure

member	type of data	describe
byRes	BYTE[]	Reserved. Maximum length is 512 bytes.

A.1.166 NET_EHOME_SS_VRB_PARAM

VRB storage information structure

member	type of data	describe
byRes	BYTE[]	Reserved. Maximum length is 512 bytes.

A.1.167 NET_EHOME_STOPPLAYBACK_PARAM

Playback stop request parameter structure

member	type of data	describe
ISessionID	LONG	The session ID for playback.
IHandle	LONG	Output parameter pointer, used as asynchronous callback in asynchronous mode Identifier, corresponding to NET_EHOME_CMSCB_DATA dwHandle
byRes	BYTE[]	Reserved. The maximum length is 120 bytes.

A.1.168 NET_EHOME_STOPSTREAM_PARAM

Preview stop request parameter structure

member	type of data	describe
ISessionID	LONG	The session ID of the preview, represented by <u>NET_ECMS_StartGetRealStreamV11</u> <u>NET_ECMS_StartGetRealStream</u> returns.
IHandle	LONG	Output parameter pointer, used as asynchronous callback in asynchronous mode Identifier, corresponding to <u>NET_EHOME_CMSCB_DATA</u> dwHandle
byRes	BYTE[]	Reserved. The maximum length is 120 bytes.

A.1.169 NET_EHOME_STOPVOICETALK_STM_PARAM

Parameter structure of voice intercom stop request through streaming media server

member	type of data	describe
ISessionID	LONG	The ID of the voice intercom session through the streaming server.
IHandle	LONG	Output parameter handle, used as asynchronous callback in asynchronous mode Identifier. Corresponds to <u>NET_EHOME_CMSCB_DATA</u> dwHandle
byRes	BYTE[]	Reserved. The maximum length is 120 bytes.

A.1.170 NET_EHOME_SWITCH_DAILY_PLAN

Daily scheduled power on/off plan parameter structure

member	type of data	describe
dwDayPlanCnt	DWORD	The number of power on and off time nodes in a day.
struDayPlan	<u>NET_EHOME_SWITCH</u> <u>PLAN[]</u>	Power on/off parameters. The maximum length is 16 bytes.

A.1.171 NET_EHOME_SWITCH_PLAN

Power on/off parameter structure

member	type of data	describe
dwId	DWORD	Configure the node ID.
dwSwitchType	DWORD	Power on/off: 1-on, 2-off.
dwSwitchDate	DWORD	Start and stop date.
dwSwitchTime	DWORD	Power on/off time, such as "235959" means the time is 23:59:59

A.1.172 NET_EHOME_SWITCH_PLAN_PARAM

The power on/off plan parameter structure sent to the terminal

member	type of data	describe
enumPlanType	<u>NET_EHOME_PLAY_SC</u> <u>HEDULE_TYPE1</u>	Plan type: weekly plan or daily plan, see enumeration.
unionSwitchPlan	Union <u>surface 16-14) Planning</u>	parameter union.

Table A-14 Planning Parameters Union

member	type of data	describe
struDailyPlan	<u>NET_EHOME_SWITCH</u> <u>DAILY_PLAN</u>	Daily planning parameters.
struWeeklyPlan	<u>NET_EHOME_SWITCH</u> <u>WEEKLY_PLAN</u>	Weekly planning parameters.

A.1.173 NET_EHOME_SWITCH_WEEKLY_PLAN

Power on/off weekly plan parameter structure

member	type of data	describe
dwWeekPlanCnt	DWORD	Days of the week for which power on/off schedules are configured.
struWeekPlan	<u>NET_EHOME_SWITCH</u> <u>DAY_OF_WEEK_PLAN</u> []	Power on/off weekly plan parameter. The maximum length is 7 bytes.

A.1.174 NET_EHOME_SWITCH_DAY_OF_WEEK_PLAN

Power on/off weekly plan parameter structure

member	type of data	describe
dwId	DWORD	Day of the week ID.
enumDayOfWeek	<u>NET_EHOME_DAY_OF</u> <u>WEEK1</u>	Week.
struDayOfWeekPlan	<u>NET_EHOME_SWITCH</u> <u>DAILY_PLAN</u>	Daily planning parameters for power on and off.

A.1.175 NET_EHOME_SYSCOMPONENT_REDUCED_ADDR

ISUPSDK (General) Development Guide

Terminal component information structure

member	type of data	describe
szComponentName BYTE[]		Component name, such as "SDK". The maximum length is 32 bytes.
szComponentPkgNam It is	BYTE[]	Component package name, such as "com.hikvision.sdk". Maximum length The length is 32 bytes.
szComponentVersion BYTE[]		Component version information, such as "v2.0.2 build 20180705". The maximum length is 32 bytes.
dwIsEnableAutoStart DWORD		Whether to start automatically at boot time.
dwIsSupportUpgrade DWORD		Whether to support remote upgrade.
byRes	BYTE[]	Reserved. The maximum length is 20 bytes.

A.1.176 NET_EHOME_TERM_DEFAULT_GATEWAY

Terminal gateway parameter structure

member	type of data	describe
szIpAddress	BYTE[]	The gateway IPv4 address. The maximum length is 32 bytes.
szIpv6Address	BYTE[]	The gateway IPv6 address. The maximum length is 32 bytes.

A.1.177 NET_EHOME_TERM_IP_ADDRESS

Terminal IP address parameter structure

member	type of data	describe
szIpVersion	BYTE[]	IP address version: v4 and v6. The maximum length is 32 bytes.
szIpAddress	BYTE[]	IP address. The maximum length is 32 bytes.
szSubnetMask	BYTE[]	Subnet mask. The maximum length is 32 bytes.

member	type of data	describe
szIpv6Address	BYTE[]	IPv6 address, currently not supported. Maximum length is 32 characters Festival.
szBitMask	BYTE[]	The gateway IPv6 address. The maximum length is 32 bytes.
strDefaultGateway <i>NET_EHOME_TERM_D</i> <i>EFAULT_GATEWAY</i>		Gateway parameters for the terminal.

A.1.178 NET_EHOME_TERM_TIME_ZONE

Terminal time zone information structure

member	type of data	describe
szTimeZone	BYTE[]	Time zone, such as GMT+08. The maximum length is 32 bytes.

A.1.179 NET_EHOME_TERMINAL_CONTROL

Terminal control parameter structure

member	type of data	describe
dwControlType	DWORD	Control command type, see enumeration <i>NET_EHOME_CONTROL_TYPE</i>
struInsertInfo	<i>NET_EHOME_INSERT_I</i> <i>NFO</i>	Insertion parameters.
struPlanCtrl	<i>NET_EHOME_PLANCTR</i> <i>LPARAM</i>	Timing plan control parameters of information release terminal.
dwConnPort	DWORD	Data transmission port, reserved.

A.1.180 NET_EHOME_TERMINAL_CONTROL_V20

Terminal control parameter structure (V20)

member	type of data	describe
dwControlType	DWORD	Control command type, see enumeration <u>NET_EHOME_CONTROL_TYPE</u>
struInsertInfo	<u>NET_EHOME_INSERT_I</u> <u>NFO_V20</u>	Insertion parameters.
struPlanCtrl	<u>NET_EHOME_PLANCTR</u> <u>LPARAM</u>	Timing plan control parameters of information release terminal.
dwConnPort	DWORD	Data transmission port, reserved.
byRes	BYTE[]	Reserved. The maximum length is 64 bytes.

A.1.181 NET_EHOME_TERMINAL_INFO

Terminal information structure

member	type of data	describe
vny <u>NET_EHOME_IDEN</u> <u>TIFI</u> <u>CATION</u>		Terminal ID.
dwNetUintType	DWORD	Network Type.
byDeviceID	BYTE[]	Device ID. The maximum length is 32 bytes (corresponding to the macro definition is "NET_EHOME_MAX_TYPE_LEN").
byPassWord	BYTE[]	Password. The maximum length is 32 bytes (the corresponding macro is defined as "NET_EHOME_MAX_TYPE_LEN").
byFirmWareVersion	BYTE[]	Firmware version information. The maximum length is 32 bytes (corresponding to macro Defined as "NET_EHOME_MAX_TYPE_LEN").
byLocalIP	BYTE[]	The communication IP address of the terminal. The maximum length is 32 bytes (for The macro should be defined as "NET_EHOME_MAX_TYPE_LEN").

ISUPSDK (General) Development Guide

member	type of data	describe
dwLocalPort	DWORD	The local port number.
dwDevType	DWORD	Equipment type.
dwManufacture	DWORD	Manufacturer: 0-Hikvision.
bDetectPackage	BOOL	Encapsulation type: 1-detection packet, 0-registration packet.
bReliableTrans	BOOL	Whether reliable transmission is supported: 1-yes, 0-no.
byServerUserName	BYTE[]	Server user name. Maximum length is 32 bytes (corresponding to macro Defined as "NET_EHOME_MAX_NAME_LEN").
byServerUserPassword	BYTE[]	Server password. Maximum length is 32 bytes (corresponding to macro definition is "NET_EHOME_MAX_NAME_LEN").
byTerminalName	BYTE[]	The terminal name to be registered. The maximum length is 32 bytes (for The macro should be defined as "NET_EHOME_MAX_NAME_ONLY".

A.1.182 NET_EHOME_TERMINAL_NAME

Terminal name structure

member	type of data	describe
szTermName	BYTE[]	Terminal name. Maximum length is 100 bytes (corresponding to macro definition is "NET_EHOME_MAX_TERM_NAME_LEN").

A.1.183 NET_EHOME_TERMINAL_PROGRESS

ISUPSDK (General) Development Guide

Terminal progress information structure

member	type of data	describe
dwTerminalId	DWORD	Terminal ID.
dwTermStatus	DWORD	Terminal status: 1-normal, 2-abnormal.
dwProgressType	DWORD	Progress type: 1-Material progress, 2-Program progress, 3-Schedule Progress, 4- Upgrade progress.
dwUniqueId	DWORD	Schedule ID, program ID, or material ID. This field is used when upgrading. invalid.
byMainProgress	BYTE	File download progress, the value range is from 0 to 100; if the value is 101, it means the progress is abnormal.
bySubProgress	BYTE	Download progress of a single file.

A.1.184 NET_EHOME_TERMINAL_REPORT_INFO

Terminal reporting data structure

member	type of data	describe
dwSubCmd	DWORD	Identifies the type of data content.
dwContentLen	DWORD	The length of the data content.
byContentBuf	BYTE[]	Buffer for storing data content. Maximum length is 2048 characters Festival.
byRes	BYTE	Reserved. The maximum length is 64 bytes.

A.1.185 NET_EHOME_TERMINAL_STATE

Terminal state structure

member	type of data	describe
enumPlayState	<u>NET_EHOME_TERMIN</u> <u>AL_PLAY_STATE</u>	Playback status.
enumTerminalType <u>NET_EHOME_TERMIN</u>	<u>TERMIN</u> <u>AL_TYPE</u>	terminal type.
enumInsertState	<u>NET_EHOME_INSERT</u> <u>STATE</u>	Interrupt status.
byVn BYTE[]		The software version number of the terminal. The maximum length is 32 bytes.
byLocalIP	BYTE[]	The IP address of the terminal. The maximum length is 32 bytes.
dwTermAbnormalStat It is	DWORD	Whether the terminal is in abnormal temperature shutdown state: 0-No, 1-yes.

A.1.186 NET_EHOME_TERMINAL_UPG_ST_REPORT

The upgrade status structure returned by the terminal

member	type of data	describe
dwLength	DWORD	The size of the structure.
dwCheckSum	DWORD	Checksum.
dwRetVal	DWORD	Status: 1000-upgrade successful, 1001-upgrading, 1002-Upgrade failed, 1003-Decoding failed, 1004-APK version is not correct Match, 1005-ROM version does not match.
dwUpgradPercent	DWORD	Upgrade progress (percentage).

A.1.187 NET_EHOME_TEXT_MESSAGE

ISUPSDK (General) Development Guide

Text message content structure

member	type of data	describe
dwId	DWORD	Text message ID.
szMsgName	BYTE[]	Message name. Maximum length is 64 bytes (corresponding to macro definition is "NET_EHOME_MAX_NAME_LEN").
szContent	BYTE[]	Message content. The maximum length is 1024 bytes (corresponding to the macro definition Meaning is "NET_EHOME_MAX_TEXT_CONTENT_ON ONE").
szBeginTime	BYTE[]	Insertion start time. The maximum length is 32 bytes (corresponding to macro Defined as "NET_EHOME_MAX_TYPE_LEN").
szEndTime	BYTE[]	Insertion end time. The maximum length is 32 bytes (corresponding to macro Defined as "NET_EHOME_MAX_TYPE_LEN").

A.1.188 NET_EHOME_TIME

Time information structure

member	type of data	describe
wYear	WORD	Year
byMonth	BYTE	month
byDay	BYTE	Day
byHour	BYTE	hour
byMinute	BYTE	point
bySecond	BYTE	Second
byRes1	BYTE	reserve.

ISUPSDK (General) Development Guide

member	type of data	describe
wMSecond	WORD	millisecond
byRes2	BYTE[]	Reserved. The maximum length is 2 bytes.

A.1.189 NET_EHOME_TIME_ADJUST_PARAM

Time synchronization parameter structure

member	type of data	describe
dwTimeZone	DWORD	Time zone, see enumeration <i>NET_EHOME_TIME_ZONE</i> .
szTime	BYTE[]	Time information, in the format of YYYY-MM-DD HH:MM:SS (year-month-day hour:minute:second). The maximum length is 32 bytes.

A.1.190 NET_EHOME_UPGRADE_CONN_PARAM

TCP link upgrade command parameter structure

member	type of data	describe
szServerIP	BYTE[]	Server IP address. The maximum length is 32 bytes.
dwServerPort	DWORD	Server port number.

A.1.191 NET_EHOME_VERSION_INFO

ISUPSDK (General) Development Guide

Version information structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
wVn BYTE[]		Software version number. Maximum length is 32 bytes (corresponding to macro definition is defined as "MAX_VERSION_LEN").
wVn BYTE[]		Encoded version number. Maximum length is 32 bytes (corresponding to macro definition is defined as "[MAX_VERSION_LEN"]).
sPanelVersion	BYTE[]	Panel version number. Maximum length is 32 bytes (corresponding to macro definition is defined as "MAX_VERSION_LEN").
sHardwareVersion	BYTE[]	Hardware version number. The maximum length is 32 bytes (corresponding to the macro definition is defined as "MAX_VERSION_LEN").
byRes	BYTE[]	Reserved. The maximum length is 124 bytes.

A.1.192 NET_EHOME_VOICE_TALK_IN

Input parameter structure of voice intercom request

member	type of data	describe
dwVoiceChan	DWORD	The channel number of the voice intercom.
struStreamSever	<u>NET_EHOME_IPADDRE</u> <u>SS</u>	The address information of the streaming media server.
byEncodingType	<u>NET_EHOME_TALK_EN</u> <u>CODING_TYPE[]</u>	Voice intercom encoding type. The maximum length is 9 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 119 bytes.

A.1.193 NET_EHOME_VOICE_TALK_OUT

Output parameter structure of voice intercom request

member	type of data	describe
ISessionID	LONG	The session ID of the voice intercom request, returned by the device.
IHandle	LONG	Output parameter handle, used as asynchronous callback in asynchronous mode Identifier. Corresponds to <u>NET_EHOME_CMSCB_DATA</u> dwHandle
byRes	BYTE[]	Reserved, set to 0. The maximum length is 124 bytes.

A.1.194 NET_EHOME_VOICETALK_DATA

Voice data structure for forwarding

member	type of data	describe
pSendBuf	BYTE*	Voice data buffer.
dwDataLen	DWORD	Voice data size.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 128 bytes.

A.1.195 NET_EHOME_VOICETALK_DATA_CB_INFO

Voice intercom callback data structure

member	type of data	describe
pData	BYTE*	Buffer for storing voice data.
dwDataLen	DWORD	The size of the buffer for storing voice data.
byRes	BYTE[]	Reserved. The maximum length is 128 bytes.

A.1.196 NET_EHOME_VOICETALK_DATA_CB_PARAM

Voice intercom data callback parameter structure

member	type of data	describe
fnVoiceTalkDataCB	<u>VOICETALK_DATA_CB</u> callback	function for voice intercom data.
pUserData	void*	User parameters.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 128 bytes.

A.1.197 NET_EHOME_VOICETALK_NEWLINK_CB_INFO

Voice intercom callback information structure (applicable to **64** -bit **Windows** or **Linux** operating system)

member	type of data	describe
szDeviceID	char[]	Device ID. The maximum length is 256 bytes (corresponding to the macro definition is "NET_EHOME_DEVICEID_LEN").
dwEncodeType	DWORD	Voice coding type: 0-G.722, 1-G.711U, 2-G.711A, 3-G.726, 4-AAC, 5-MP2L2, 6-PCM
sDeviceSerial	char[]	Device serial number. Maximum length is 12 bytes (corresponding to macro definition is defined as "NET_EHOME_SERIAL_LEN").
dwAudioChan	DWORD	Voice intercom channel number.
lSessionID	LONG	The session ID of the voice intercom, returned by the device.
byToken	BYTE[]	Token. Maximum length is 64 bytes.
fnVoiceTalkDataCB	<u>VOICETALK_DATA_CB</u> data	callback function, its size is 8 bytes.
pUserData	void*	User parameters, its size is 8 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 48 bytes.

ISUPSDK (General) Development Guide

Voice intercom callback information structure (applicable to **32-bit Windows** or **Linux** operating system)

member	type of data	describe
szDeviceID	char[]	Device ID. The maximum length is 256 bytes (corresponding to the macro definition is "NET_EHOME_DEVICEID_LEN").
dwEncodeType	DWORD	Voice coding type: 0-G.722, 1-G.711U, 2-G.711A, 3-G.726, 4-AAC, 5-MP2L2, 6-PCM
sDeviceSerial	char[]	Device serial number. Maximum length is 12 bytes (corresponding to macro definition is defined as "NET_EHOME_SERIAL_LEN").
dwAudioChan	DWORD	Voice intercom channel number.
lSessionID	LONG	The session ID of the voice intercom, returned by the device.
byToken	BYTE[]	Token. Maximum length is 64 bytes.
fnVoiceTalkDataCB	<u>VOICETALK_DATA_CB</u> .data_callback	callback function, its size is 4 bytes.
byRes1	BYTE[]	Reserved, the maximum length is 4 bytes.
pUserData	void*	User parameters, its size is 4 bytes.
byRes2	BYTE[]	Reserved, the maximum length is 4 bytes.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 48 bytes.

A.1.198 NET_EHOME_VOICETALK_PARA

Voice intercom or voice forwarding parameter structure

member	type of data	describe
bNeedCBNoEncData BOOL		The callback voice type: 0-encoded voice, 1-encoded voice Voice (not supported during voice forwarding).
cbVoiceDataCallBack <u>fVoiceDataCallBack</u>		callback function for audio data.

member	type of data	describe
dwEncodeType	DWORD	Supported voice coding types: 0-G.722, 1-G.711U, 2-G.711Aÿ3-G.726ÿ4-AACÿ5-MP2L2ÿ6-PCMÿ
navel	void*	User parameters.
byVoiceTalk	BYTE	Working mode: 0-voice intercom, 1-voice forwarding.
byDevAudioEnc	BYTE	Audio coding type of the device: 0-G.722, 1-G.711U, 2-G.711Aÿ3-G.726ÿ4-AACÿ5-MP2L2ÿ6-PCMÿ
byRes1	BYTE[]	Reserved, set to 0. The maximum length is 2 bytes.
IHandle	LONG	Message handle for voice intercom asynchronous callback.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 56 bytes.

A.1.199 NET_EHOME_VOLUME_DAILY_PLAN

Volume adjustment daily plan structure

member	type of data	describe
dwDayPlanCnt	DWORD	The number of time points during the day when the volume is adjusted.
struDayPlan	<u>NET_EHOME_VOLUME</u> <u>PLAN</u>	Volume adjustment daily plan parameters. The maximum length is 8 bytes.

A.1.200 NET_EHOME_VOLUME_DAY_OF_WEEK_PLAN

Volume adjustment weekly plan parameter structure

member	type of data	describe
dwId	DWORD	Day of the week ID.
enumDayOfWeek	<u>NET_EHOME_DAY_OF_WEEK1</u>	Week.
struDayOfWeekPlan	<u>NET_EHOME_VOLUME_DAILY_PLAN</u>	Volume adjustment daily plan parameters.

A.1.201 NET_EHOME_VOLUME_PLAN

Volume parameter structure

member	type of data	describe
dwId	DWORD	Volume adjustment ID.
dwVolumeValue	DWORD	Volume value.
dwVolumeBeginDate DWORD		The volume adjustment start date.
dwVolumeBeginTime DWORD		Volume adjustment start time, such as "235959" indicates time 23:59:59
dwVolumeEndDate DWORD		The volume leveling end date.
dwVolumeEndTime DWORD		Volume adjustment end time.

A.1.202 NET_EHOME_VOLUME_PLAN_PARAM

The volume adjustment parameter structure to be sent to the terminal

member	type of data	describe
enumPlanType	<u>NET_EHOME_PLAY_SC</u> <u>HEDULE_TYPE1</u>	Plan type.
unionVolumePlan	Union $\ddot{\vee}$ <u>surface 16-15) Volume</u>	Regulation Program Consortium.

Table A-15 Volume Control Plan Consortium

member	type of data	describe
struDailyPlan	<u>NET_EHOME_VOLUME</u> <u>DAILY_PLAN</u>	Daily plan.
struWeeklyPlan	<u>NET_EHOME_VOLUME</u> <u>WEEKLY_PLAN</u>	weekly plan.

A.1.203 NET_EHOME_VOLUME_WEEKLY_PLAN

Volume adjustment weekly plan parameter structure

member	type of data	describe
dwWeekPlanCnt	DWORD	The days of the week when there are volume adjustment tasks.
struWeekPlan	<u>NET_EHOME_VOLUME</u> <u>DAY_OF_WEEK_PLAN</u> []	Volume adjustment weekly plan parameters. The maximum length is 7 bytes.

A.1.204 NET_EHOME_WEATHER_INFO

Weather information structure

member	type of data	describe
byCityCode	BYTE[]	City code. Maximum length is 32 bytes.
byWeatherData	BYTE[]	Weather data. The maximum length is 4*1024 bytes.

A.1.205 NET_EHOME_WIRELESS_INFO_CFG

Wireless parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
dwInfoTransInterval DWORD		Upload interval, unit: seconds.
byEnable	BYTE	Supported: 0-No, 1-Yes.
byRes	BYTE[]	Reserved. The maximum length is 47 bytes.

A.1.206 NET_EHOME_XML_CFG

Transmission configuration parameter structure

member	type of data	describe
pCmdBuf	void*	Command in string format.
dwCmdLen	DWORD	The number of characters in the command.
pInBuf	void*	Input parameters.
dwInSize	DWORD	Enter the buffer size.
pOutBuf	void*	Output parameters.
dwOutSize	DWORD	Output buffer size.
dwSendTimeOut	DWORD	Data sending timeout, unit: milliseconds, default: 5s.

member	type of data	describe
dwRecvTimeOut	DWORD	Data receiving timeout, unit: milliseconds, default: 5s.
pStatusBuf	void*	A buffer to store the return status information (in XML format); if not required, it can be set to "empty".
dwStatusSize	DWORD	The size of the status buffer.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 24 bytes.

Related interfaces

[NCMMCn](#)

A.1.207 NET_EHOME_XML_REMOTE_CTRL_PARAM

Transmission parameter structure

member	type of data	describe
dwSize	DWORD	The size of the structure.
nb	void*	Buffer for storing input parameters, controlled by the transmitted ISUP Command decision.
wnB	DWORD	Enter the buffer size.
dwSendTimeOut	DWORD	Data sending timeout, unit: milliseconds, default: 5 s.
dwRecvTimeOut	DWORD	Data receiving timeout, unit: milliseconds, default: 5 s.
B	void*	Output buffer.
wB	DWORD	Output buffer size.
B	void*	Status buffer. If this parameter is not needed, it can be set to "empty".
wB DWORD		The size of the status buffer.
byRes	BYTE[]	Reserved, set to 0. The maximum length is 16 bytes.

Related interfaces

[NET_ECMS_XMLRemoteControl](#)

A.1.208 NET_EHOME_ZONE

Region parameter structure

member	type of data	describe
dwX	DWORD	X-axis coordinate.
dwY	DWORD	Y-axis coordinate.
dwWidth	DWORD	Width of the region.
dwHeight	DWORD	Area height.

A.2 Enumeration Definition

A.2.1 EN_ALARM_TYPE

Enumeration alarm type

Enumeration definition

```
typedef enum{
    ALARM_TYPE_DISK_FULL = 0,
    ALARM_TYPE_DISK_WRERROR = 1,
    ALARM_TYPE_VIDEO_LOST = 5,
    ALARM_TYPE_EXTERNAL = 6,
    ALARM_TYPE_VIDEO_COVERED = 7,
    ALARM_TYPE_MOTION = 8,
    ALARM_TYPE_STANDARD_NOTMATCH = 9,
    ALARM_TYPE_SPEEDLIMIT_EXCEED = 10,
    ALARM_TYPE_PIR = 11,
    ALARM_TYPE_WIRELESS = 12,
    ALARM_TYPE_CALL_HELP = 13,
    ALARM_TYPE_DISARM = 14,
    ALARM_TYPE_STREAM_PRIVATE = 15,
    ALARM_TYPE_PIC_UPLOAD_FAIL = 16,
    ALARM_TYPE_LOCAL_REC_EXCEPTION = 17,
    ALARM_TYPE_UPGRADE_FAIL = 18,
    ALARM_TYPE_ILLEGAL_ACCESS = 19,
```

```
ALARM_TYPE_SOUNDLIMIT_EXCEED = 80,  
ALARM_TYPE_TRAFFIC_VIOLATION = 90,  
ALARM_TYPE_ALARM_CONTROL = 96,  
ALARM_TYPE_FACE_DETECTION = 97,  
ALARM_TYPE_DEFUSE_DETECTION = 98,  
ALARM_TYPE_AUDIO_EXCEPTION = 99,  
ALARM_TYPE_SCENE_CHANGE = 100,  
ALARM_TYPE_TRAVERSE_PLANE = 101,  
ALARM_TYPE_ENTER_AREA = 102,  
ALARM_TYPE_LEAVE_AREA = 103,  
ALARM_TYPE_INTRUSION = 104,  
ALARM_TYPE_LOITER = 105,  
ALARM_TYPE_LEFT_TAKE = 106,  
ALARM_TYPE_CAR_STOP = 107,  
ALARM_TYPE_MOVE_FAST = 108,  
ALARM_TYPE_HIGH_DENSITY = 109,  
ALARM_TYPE_PDC_BY_TIME = 110,  
ALARM_TYPE_PDC_BY_FRAME = 111,  
ALARM_TYPE_LEFT = 112,  
ALARM_TYPE_TAKE = 113,  
ALARM_TYPE_ROLLOVER = 114,  
ALARM_TYPE_COLLISION = 115,  
ALARM_TYPE_FLOW_OVERRUN = 256,  
ALARM_TYPE_WARN_FLOW_OVERRUN = 257,  
ALARM_TYPE_DEV_CHANGED_STATUS = 700,  
ALARM_TYPE_CHAN_CHANGED_STATUS = 701,  
ALARM_TYPE_HD_CHANGED_STATUS = 702,  
ALARM_TYPE_DEV_TIMING_STATUS = 703,  
ALARM_TYPE_CHAN_TIMING_STATUS = 704,  
ALARM_TYPE_HD_TIMING_STATUS = 705,  
ALARM_TYPE_RECORD_ABNORMAL = 706,  
ALARM_TYPE_ENV_LIMIT = 8800,  
ALARM_TYPE_ENV_REAL_TIME = 8801,  
ALARM_TYPE_ENV_EXCEPTION = 8802,  
ALARM_TYPE_HIGH_TEMP = 40961,  
ALARM_TYPE_ACC_EXCEPTION = 40962,  
ALARM_TYPE_RAPID_ACCELERATION = 40963,  
ALARM_TYPE_RAPID_DECELERATION = 40964,  
ALARM_TYPE_COLLISION_V40 = 40965,  
ALARM_TYPE_ROLLOVER_V40 = 40966,  
ALARM_TYPE_RAPID_TURN_LEFT = 40967,  
ALARM_TYPE_RAPID_TURN_RIGHT = 40968,  
ALARM_TYPE_ABNORMAL_DRIVING_BEHAVIOR = 40969,  
ALARM_TYPE_OVERLOAD = 40970,  
ALARM_TYPE_LEFT_CROSS_LINE = 40971,  
ALARM_TYPE_RIGHT_CROSS_LINE = 40972,  
ALARM_TYPE_OPEN_DOOR_WITH_SPEED = 40973,  
ALARM_TYPE_ADAS = 40974,  
ALARM_TYPE_RADAR = 41009  
}EN_ALARM_TYPE;
```

member

ALARM_TYPE_DISK_FULL

Hard disk full alarm

ALARM_TYPE_DISK_WRERROR

Hard disk read and write error alarm

ALARM_TYPE_VIDEO_LOST

Video (signal) loss alarm

ALARM_TYPE_EXTERNAL

External (semaphore) alarm

ALARM_TYPE_VIDEO_COVERED

Video Cover Alarm

ALARM_TYPE_MOTION

Motion Detection

ALARM_TYPE_STANDARD_NOTMATCH

Video format mismatch alarm

ALARM_TYPE_SPEEDLIMIT_EXCEED

Overspeed alarm

ALARM_TYPE_PIR

PIR Alarm

ALARM_TYPE_WIRELESS

Wireless alarm

ALARM_TYPE_CALL_HELP

Call for help

ALARM_TYPE_DISARM

Arm and disarm alarm

ALARM_TYPE_STREAM_PRIVATE

Stream privacy status change alarm

ALARM_TYPE_PIC_UPLOAD_FAIL

Device upload picture failure alarm

ALARM_TYPE_LOCAL_REC_EXCEPTION

Device local video (forensics) abnormal alarm

ALARM_TYPE_UPGRADE_FAIL

Device version upgrade failure alarm

ALARM_TYPE_ILLEGAL_ACCESS

Illegal access alarm

ALARM_TYPE_SOUNDLIMIT_EXCEED

Alarm when the decibel level exceeds the limit

ALARM_TYPE_TRAFFIC_VIOLATION

Traffic violation alarm

ALARM_TYPE_ALARM_CONTROL

Arming alarm

ALARM_TYPE_FACE_DETECTION

Face detection alarm

ALARM_TYPE_DEFOCUS_DETECTION

Out-of-focus detection alarm

ALARM_TYPE_AUDIO_EXCEPTION

Abnormal audio alarm

ALARM_TYPE_SCENE_CHANGE

Scene change detection alarm

ALARM_TYPE_TRAVERSE_PLANE

Cross-border detection alarm

ALARM_TYPE_ENTER_AREA

Entering area detection alarm

ALARM_TYPE_LEAVE_AREA

Leaving area detection alarm

ALARM_TYPE_INTRUSION

Area intrusion detection alarm

ALARM_TYPE_LOITER

Loitering detection alarm

ALARM_TYPE_LEFT_TAKE

Left-behind item removal detection alarm

ALARM_TYPE_CAR_STOP

Parking detection alarm

ALARM_TYPE_MOVE_FAST

Fast motion detection alarm

ALARM_TYPE_HIGH_DENSITY

Crowd detection alarm

ALARM_TYPE_PDC_BY_TIME

Passenger flow statistics alarm by time period

ALARM_TYPE_PDC_BY_FRAME

Single frame passenger flow statistics alarm

ALARM_TYPE_LEFT

Object left behind detection alarm

ALARM_TYPE_TAKE

Object removal detection alarm

ALARM_TYPE_ROLLOVER

Rollover alarm

ALARM_TYPE_COLLISION

Collision Alarm

ALARM_TYPE_FLOW_OVERRUN

Flow rate over limit alarm

ALARM_TYPE_WARN_FLOW_OVERRUN

Personnel over limit reminder

ALARM_TYPE_DEV_CHANGED_STATUS

Device status change alarm

ALARM_TYPE_CHAN_CHANGED_STATUS

Channel status change alarm

ALARM_TYPE_HD_CHANGED_STATUS

Hard disk status change alarm

ALARM_TYPE_DEV_TIMING_STATUS

Scheduled upload of device status alarm

ALARM_TYPE_CHAN_TIMING_STATUS

Scheduled upload channel status alarm

ALARM_TYPE_HD_TIMING_STATUS

Scheduled upload of hard disk status alarm

ALARM_TYPE_RECORD_ANORMAL

Abnormal video alarm

ALARM_TYPE_ENV_LIMIT

Dynamic environment quantity over-limit alarm

ALARM_TYPE_ENV_REAL_TIME

Real-time data upload and alarm of dynamic environment

ALARM_TYPE_ENV_EXCEPTION

Abnormal dynamic environment quantity upload alarm

ALARM_TYPE_HIGH_TEMP

Over temperature alarm

ALARM_TYPE_ACC_EXCEPTION

Acceleration abnormal alarm

ALARM_TYPE_RAPID_ACCELERATION

Rapid acceleration alarm

ALARM_TYPE_RAPID_DECELERATION

Rapid deceleration alarm

ALARM_TYPE_COLLISION_V40

Collision Alarm

ALARM_TYPE_ROLLOVER_V40

Rollover alarm

ALARM_TYPE_RAPID_TURN_LEFT

Sharp left turn alarm

ALARM_TYPE_RAPID_TURN_RIGHT

Sharp right turn alarm

ALARM_TYPE_ANORMAL_DRIVING_BEHAVIOR

Abnormal driving behavior alarm

ALARM_TYPE_OVERLOAD

Overload alarm

ALARM_TYPE_LEFT_CROSS_LINE

Left pressure line alarm

ALARM_TYPE_RIGHT_CROSS_LINE

Right pressure line alarm

ALARM_TYPE_OPEN_DOOR_WITH_SPEED

With speed door opening alarm

ALARM_TYPE_ADAS

Active safety (ADAS) warning

ALARM_TYPE_RADAR

Radar alarm

A.2.2 LONG_LINK_MSG

Enumerate long connection types

Enumeration definition

```
typedef enum{
    LONG_CFG_CREATED =0,
    LONG_CFG_CREATE_FAIL =1,
    LONG_CFG_DATA =2,
    LONG_CFG_TERMINATE =3
}LONG_LINK_MSG;
```

member

LONG_CFG_CREATED

The long connection is established successfully.

LONG_CFG_CREATE_FAIL

Failed to establish a persistent connection.

LONG_CFG_DATA

Long connection normal data.

LONG_CFG_TERMINATE

Destroy the persistent connection.

A.2.3 NET_CMS_ENUM_PROXY_TYPE

Enumeration proxy type

Enumeration definition

```
typedef enum tagNET_CMS_ENUM_PROXY_TYPE{
    ENUM_PROXY_TYPE_NETSDK = 0,
    ENUM_PROXY_TYPE_HTTP = 1
}NET_CMS_ENUM_PROXY_TYPE;
```

member

ENUM_PROXY_TYPE_NETSDK

Hikvision private agreement agent.

ENUM_PROXY_TYPE_HTTP

HTTP proxy.

A.2.4 NET_EHOME_ADDRESS_TYPE

Enumeration of address types.

Enumeration definition

```
typedef enum{
    NET_EHOME_HOST_MODE,
    NET_EHOME_IP_MODE,
}NET_EHOME_ADDRESS_TYPE;
```

member

NET_EHOME_HOST_MODE

domain name.

NET_EHOME_IP_MODE

IP address.

A.2.5 NET_EHOME_ADDRFORMAT_CMD

Enumeration of address types.

Enumeration definition

```
typedef enum{
    ADDRTYPE_IPV4 = 0,
    ADDRTYPE_IPV6 = 1,
    HOSTNAME = 2,
}NET_EHOME_ADDRFORMAT_CMD;
```

member

ADDRTYPE_IPV4

IPv4 address.

ADDRTYPE_IPV6

IPv6 address.

HOSTNAME

domain name.

A.2.6 NET_EHOME_CALLBACK_TYPE

Enumeration callback type.

Enumeration definition

```
typedef enum tagNET_EHOME_CALLBACK_TYPE{
    NET_EHOME_REALSTREAM_CB = 0,
    NET_EHOME_PLAYBACK_CB = 1,
    NET_EHOME_VOICETALK_CB = 2
    NET_EHOME_PASSTHROUGH_CB = 3
    NET_EHOME_V2_VOICETALK_CB = 4
}NET_EHOME_CALLBACK_TYPE;
```

member

NET_EHOME_REALSTREAM_CB

Preview asynchronous callback, corresponding to NET_ECMS_StartGetRealStream , NET_ECMS_StartGetRealStreamV11 ,

NET_ECMS_StartPushRealStream ѕ NET_ECMS_StopGetRealStream ѕ

Asynchronous return of the NET_ECMS_StopGetRealStreamEx interface.

NET_EHOME_PLAYBACK_CB

Playback asynchronous callback, corresponding to NET_ECMS_StartPlayBack , NET_ECMS_StopPlayBack ,

NET_ECMS_StopPlayBackEx ѕ NET_ECMS_StartPushPlayBack NET_ECMS_StartPushPlayBack ѕ

Asynchronous return of the NET_ECMS_PlayBackOperate interface.

NET_EHOME_VOICETALK_CB

Voice (intercom) forwarding asynchronous callback. Corresponding to NET_ECMS_StartVoiceWithStmServer ,

NET_ECMS_StartPushVoiceStream ѕ NET_ECMS_StopVoiceTalkWithStmServer ѕ

Asynchronous return of the NET_ECMS_StopVoiceTalkWithStmServerEx interface.

NET_EHOME_PASSTHROUGH_CB

ISAPI pass-through asynchronous callback. Corresponding to the asynchronous return of the *NET_ECMS_ISAPIPassThrough* interface.

NET_EHOME_V2_VOICETALK_CB

ISUP2.0 protocol voice intercom asynchronous callback. Corresponding to the asynchronous return of the *NET_ECMS_StartVoiceTalk* interface.

A.2.7 NET_EHOME_CMS_INIT_CFG_TYPE

Enumeration initialization configuration type.

Enumeration definition

```
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;
```

member

NET_EHOME_CMS_INIT_CFG_LIBEAY_PATH

Set the path to the OpenSSL encryption library (libeay32.dll or libcrypto.so). The path is the full path including the library name.

NET_EHOME_CMS_INIT_CSSLEAY_PATH

Set the path where the OpenSSL communication library (ssleay32.dll or libssl.so) is located. The path is the full path including the library name.

A.2.8 NET_EHOME_CONTROL_TYPE

Enumerate terminal control commands

Enumeration definition

```
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, };
```

member

NET_EHOME_CONTROL_STARTPLAY

Start playing.

NET_EHOME_CONTROL_STOPPLAY

Stop play.

NET_EHOME_CONTROL_INSERT

Start inserting.

NET_EHOME_CONTROL_STOPINSERT

Stop interrupting.

NET_EHOME_CONTROL_POWERON

Turn on the computer.

NET_EHOME_CONTROL_POWEROFF

Power off.

NET_EHOME_CONTROL_REBOOT

Reboot.

NET_EHOME_CONTROL_RESTORECONFIG

Restore default settings.

NET_EHOME_CONTROL_SCREENOPEN

Screen on.

NET_EHOME_CONTROL_SCREENCLOSE

Screen off.

NET_EHOME_CONTROL_PLANCTRL

Scheduled control of volume and power on/off schedule.

NET_EHOME_CONTROL_CANCELINSERT

Cancel the interruption.

NET_EHOME_CONTROL_PUBLISH_STATE_SHOW

Displays the progress of publishing.

NET_EHOME_CONTROL_PUBLISH_STATE_HIDE

Hide post progress.

A.2.9 NET_EHOME_DAY_OF_WEEK1

Enumerate the days of the week.

Enumeration definition

```
typedef enum{
    INVALID_DAY = 0,
    MONDAY = 1,
    TUESDAY = 2,
    WEDNESDAY = 3,
    THURSDAY = 4,
    FRIDAY = 5,
    SATURDAY = 6,
    SUNDAY = 7,
}NET_EHOME_DAY_OF_WEEK1;
```

member

INVALID_DAY

invalid

MONDAY

Monday

TUESDAY

Tuesday

WEDNESDAY

Wednesday

THURSDAY

Thursday

FRIDAY

Friday

SATURDAY

Saturday

SUNDAY

Sunday

A.2.10 NET_EHOME_EALARM_INIT_CFG_TYPE

Enumerates the initialization configuration types of the alarm management server.

Enumeration definition

```
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;
```

member

NET_EHOME_EALARM_INIT_CFG_LIBEAY_PATH

Set the path to the OpenSSL encryption library (libeay32.dll or libcrypto.so). The path is the full path including the library name.

NET_EHOME_EALARM_INIT_CFG_SSLEAY_PATH

Set the path where the OpenSSL communication library (ssleay32.dll or libssl.so) is located. The path is the full path including the library name.

A.2.11 NET_EHOME_ERROR_CODE

Enumerates server and endpoint defined error types.

Enumeration definition

```
enum NET_EHOME_ERROR_CODE{
    EHOME_OK = 0,
    EHOME_PASSEWORD_ERROR = 1,
    EHOME_NOT_SUPPORT = 2,
    EHOME_USERNAME_ERROR = 3,
    EHOME_USERNAME_PASSWORD_ERROR = 4,
    EHOME_TERM_RNAME_REPEAT = 5,
    EHOME_TERM_SERIAL_REPEAT = 6,
    EHOME_PARAM_ERROR ); = 7,
```

member

EHOME_OK

There are no errors.

EHOME_PASSEWORD_ERROR

wrong password.

EHOME_NOT_SUPPORT

The terminal device is not supported.

EHOME_USERNAME_ERROR

username error.

EHOME_USERNAME_PASSWORD_ERROR

The username and password are incorrect.

EHOME_TERM_RNAME_REPEAT

The terminal name is duplicated.

EHOME_TERM_SERIAL_REPEAT

The terminal serial number is duplicated.

EHOME_PARAM_ERROR

Parameter error.

A.2.12 NET_EHOME_ESTREAM_INIT_CFG_TYPE

Enumerates the initialization configuration types of the streaming media server.

Enumeration definition

```
typedef enum tagNET_EHOME_ESTREAM_INIT_CFG_TYPE{ = 0,
    NET_EHOME_ESTREAM_INIT_CFG_LIBEAY_PATH = 1,
    NET_EHOME_ESTREAM_INIT_CFG_SSLEAY_PATH
    NET_EHOME_ESTREAM_INIT_CFG_USERCERTIFICATE_PATH = 2,
    NET_EHOME_ESTREAM_INIT_CFG_USERPRIVATEKEY_PATH = 3
)NET_EHOME_ESTREAM_INIT_CFG_TYPE;
```

member

NET_EHOME_ESTREAM_INIT_CFG_LIBEAY_PATH

Set the path to the OpenSSL encryption library (ie, libeay32.dll or libcrypto.so). The path must be a full path including the library name.

NET_EHOME_ESTREAM_INIT_CFG_SSLEAY_PATH

Set the path to the OpenSSL communication library (ie, ssleay32.dll or libssl.so). The path must be a full path including the library name.

NET_EHOME_ESTREAM_INIT_CFG_USERCERTIFICATE_PATH

Set the server certificate path for TLS monitoring and DTLS monitoring (TLS and DTLS reuse the same certificate). If the encryption certificate is not set, the streaming interface will report an error in the playback, preview, and voice forwarding TCP encryption services.

NET_EHOME_ESTREAM_INIT_CFG_USERPRIVATEKEY_PATH

Set the private key certificate path for TLS monitoring and DTLS monitoring (TLS and DTLS reuse the same certificate). If the encryption certificate is not set, the streaming interface will report an error in the playback, preview, and voice forwarding TCP encryption services.

A.2.13 NET_EHOME_INSERT_STATE

Enumerates the insertion status.

Enumeration definition

```
typedef enum{
    NET_EHOME_INVALID_INSERT_TYPE = 0,
    NET_EHOME_INSERT_END         = 1,
    NET_EHOME_INSERT_PUBLISH     = 2,
    NET_EHOME_INSERT_START       = 3,
    NET_EHOME_INSERT_FAIL        = 4,
}NET_EHOME_INSERT_STATE;
```

member

NET_EHOME_INVALID_INSERT_TYPE

Invalid type.

NET_EHOME_INSERT_END

The interruption ends.

NET_EHOME_INSERT_PUBLISH

Send a data packet to the terminal.

NET_EHOME_INSERT_START

The interlude begins.

NET_EHOME_INSERT_FAIL

Insertion failed.

A.2.14 NET_EHOME_IPC_TYPE

Enumerates the types of network cameras.

Enumeration definition

```
typedef enum{
    NET_EHOME_NORMAL_IPC,
    NET_EHOME_SMART_PASSENGER_IPC,
    NET_EHOME_SMART_VALID_PASSENGER_IPC,
}NET_EHOME_IPC_TYPE;
```

member

NET_EHOME_NORMAL_IPC

Ordinary network camera.

NET_EHOME_SMART_PASSENGER_IPC

People counting network camera.

NET_EHOME_SMART_VALID_PASSENGER_IPC

Image recognition network camera.

A.2.15 NET_EHOME_LOCAL_CFG_TYPE

Enumerates local configuration types.

Enumeration definition

```
enum NET_EHOME_LOCAL_CFG_TYPE{
    UNDEFINE          = -1,
    ACTIVE_ACCESS_SECURITY = 0,
    AMS_ADDRESS = 1,
    SEND_PARAM = 2,
    SET_REREGISTER_MODE      = 3,
    LOCAL_CFG_TYPE_GENERAL = 4,
    COM_PATH = 5,
    SESSIONKEY_REQ_MOD      = 6
    DEV_DAS_PINGREO_CALLBACK = 7,
    REGISTER_LISTEN_MODE     = 8,
    STREAM_PLAYBACK_PARAM = 9
}NET_EHOME_LOCAL_CFG_TYPE, *LPNET_EHOME_LOCAL_CFG_TYPE;
```

member

UNDEFINE

Undefined.

ACTIVE_ACCESS_SECURITY

Set security parameters for devices connecting to the server.

AMS_ADDRESS

Set the local loopback address information of the Alarm Management Server (AMS).

SEND_PARAM

Set the sending parameters.

SET_REREGISTER_MODE

Set the re-registration mode for the device.

LOCAL_CFG_TYPE_GENERAL

Set common parameters.

COM_PATH

Configure the COM path.

SESSIONKEY_REQ_MOD

Sets the callback status for session key requests.

DEV_DAS_PINGREO_CALLBACK

Device heartbeat registration callback.

REGISTER_LISTEN_MODE

Register listening mode, the corresponding structure is *NET_EHOME_REGISTER_LISTEN_MODE*.

STREAM_PLAYBACK_PARAM

Replay local parameter configuration.

A.2.16 NET_EHOME_PLANCTRL_TYPE

The type of enumeration control.

Enumeration definition

```
enum NET_EHOME_PLANCTRL_TYPE{
    NET_EHOME_CONTROL_SWITCHPLAY = 1,
    NET_EHOME_CONTROL_VOLUMEPLAY = 2,
    NET_EHOME_CONTROL_INPUTPLAY =
3,};
```

member

NET_EHOME_CONTROL_SWITCHPLAY

Scheduled power on and off plan.

NET_EHOME_CONTROL_VOLUMEPLAY

Volume adjustment.

NET_EHOME_CONTROL_INPUTPLAY

Whether to enable the scheduled input schedule.

A.2.17 NET_EHOME_PLAY_SCHEDULE_TYPE1

Enumeration of plan types.

Enumeration definition

```
typedef enum{
    INVALID_PLAN = 0,
    DAYLY_PLAN = 1,
    WEEKLY_PLAN = 2,
    SELF_DEF_PLAN = 3,
    CIRCLE_PLAN = 4,
}NET_EHOME_PLAY_SCHEDULE_TYPE1;
```

member

INVALID_PLAN

Invalid schedule type.

DAYLY_PLAN

Daily plan.

WEEKLY_PLAN

weekly plan.

SELF_DEF_PLAN

Custom plan.

CIRCLE_PLAN

Carousel plan.

A.2.18 NET_EHOME_PLAYBACK_OPERATE_MODE

Enumerates the control operation types.

Enumeration definition

```
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;
```

member

PLAYBACK_OPERATE_UNKNOW

Invalid control action type.

PLAYBACK_OPERATE_PAUSE

Pause playback.

PLAYBACK_OPERATE_RESTART

Resume playback.

A.2.19 NET_EHOME_POS_MODE

Enumeration of coordinate types.

Enumeration definition

```
enum NET_EHOME_POS_MODE{
    NET_EHOME_POS_MODE_ILLEGAL = 0,
    NET_EHOME_POS_MODE_STANDARD = 1,
    NET_EHOME_POS_MODE_RESOLUTION = 2,};
```

member

NET_EHOME_POS_MODE_ILLEGAL

Invalid coordinate type.

NET_EHOME_POS_MODE_STANDARD

Base coordinates, such as 1920 × 1920.

NET_EHOME_POS_MODE_RESOLUTION

Resolution coordinates.

A.2.20 NET_EHOME_POWER_ONOFF

Enumerates the power on/off status.

Enumeration definition

```
typedef enum{
    POWER_ON = 1,
    POWER_OFF = 2,
}NET_EHOME_POWER_ONOFF;
```

member

POWER_ON

Turn on the computer.

POWER_OFF

Power off.

A.2.21 NET_EHOME_PROGRAM_INSERT_TYPE

Enumeration of interstitial types.

Enumeration definition

```
enum NET_EHOME_PROGRAM_INSERT_TYPE{
    NET_EHOME_INSERT_TYPE_MATERIAL = 1,
    NET_EHOME_INSERT_TYPE_PROGRAM = 2,};
```

member

NET_EHOME_INSERT_TYPE_MATERIAL

material.

NET_EHOME_INSERT_TYPE_PROGRAM

programme.

A.2.22 NET_EHOME_REFRESH_TYPE

Enumerates the data refresh types.

Enumeration definition

```
typedef enum{
    NET_EHOME_SINGLE_REFRESH_TYPE = 1,
    NET_EHOME_WHOLE_REFRESH_TYPE = 2,
    NET_EHOME_OTHER_REFRESH_TYPE = 3,
}NET_EHOME_REFRESH_TYPE;
```

member

NET_EHOME_SINGLE_REFRESH_TYPE

Refresh single data.

NET_EHOME_WHOLE_REFRESH_TYPE

Refresh all data.

NET_EHOME_OTHER_REFRESH_TYPE

other.

A.2.23 NET_EHOME_REGISTER_TYPE

Enumeration of registered callback types.

Enumeration definition

```
typedef enum tagNET_EHOME_REGISTER_TYPE{
    ENUM_UNKNOWN =-1,
    ENUM_DEV_ON =0,
    ENUM_DEV_OFF =1,
    ENUM_DEV_ADDRESS_CHANGED =2,
    ENUM_EHOME50_DEV_AUTH =3,
    ENUM_EHOME50_DEV_SESSIONKEY =4,
    ENUM_EHOME50_DEV_DAS_REQ =5,
    ENUM_DEV_SESSIONKEY_REQ =6,
    ENUM_DEV_DAS_REREGISTER =7,
    ENUM_DEV_DAS_PINGREO =8
    ENUM_DEV_DAS_EHOMEKEY_ERROR =9,
    ENUM_DEV_SESSIONKEY_ERROR =10,
    ENUM_DEV_SLEEP =11
}NET_EHOME_REGISTER_TYPE
```

member

ENUM_UNKNOWN

Unknown information.

ENUM_DEV_ON

Device online information.

ENUM_DEV_OFF

Device offline information.

ENUM_DEV_ADDRESS_CHANGED

Device address change information.

ENUM_EHOME50_DEV_AUTH

Supports authentication information for ISUP version 5.0 devices.

ENUM_EHOME50_DEV_SESSIONKEY

Session key information for devices supporting ISUP version 5.0.

ENUM_EHOME50_DEV_DAS_REQ

Supports redirection request information for version 5.0 ISUP devices.

ENUM_DEV_SESSIONKEY_REQ

Supports SessionKey request information for ISUP version 5.0 devices.

ENUM_DEV_DAS_REREGISTER

Device re-registration information.

ENUM_DEV_DAS_PINGREO

Device registration heartbeat.

ENUM_DEV_DAS_EHOMEKEY_ERROR

Password verification failure message.

ENUM_DEV_SESSIONKEY_ERROR

Sessionkey interaction exception.

ENUM_DEV_SLEEP

The device goes into sleep mode



illustrate

In dormant mode, the device cannot respond to requests for preview, playback, voice intercom, configuration, etc.; the interface can be called

[NET_ECMS_WakeUp](#) wakes up the device

See Also

[NET_ECMS_WakeUp](#)

A.2.24 NET_EHOME_SS_CLIENT_TYPE

Enumerates file upload client types.

Enumeration definition

```
enum NET_EHOME_SS_CLIENT_TYPE(
    NET_EHOME_SS_CLIENT_TYPE_TOMCAT = 1,
    NET_EHOME_SS_CLIENT_TYPE_VRB = 2,
    NET_EHOME_SS_CLIENT_TYPE_KMS = 3,
    NET_EHOME_SS_CLIENT_TYPE_CLOUD = 4,
    NET_EHOME_SS_CLIENT_TYPE_CENTRAL= 5
)NET_EHOME_SS_CLIENT_TYPE
```

member

NET_EHOME_SS_CLIENT_TYPE_TOMCAT

Client that supports Tomcat protocol

NET_EHOME_SS_CLIENT_TYPE_VRB

Clients supporting VRB protocol

NET_EHOME_SS_CLIENT_TYPE_KMS

Clients that support the KMS protocol

NET_EHOME_SS_CLIENT_TYPE_CLOUD

Clients that support ISUP5.0

NET_EHOME_SS_CLIENT_TYPE_CENTRAL

Central storage streaming storage protocol

A.2.25 NET_EHOME_SS_INIT_CFG_TYPE

Enumeration initialization configuration type.

Enumeration definition

```
enum NET_EHOME_SS_INIT_CFG_TYPE{
    NET_EHOME_SS_INIT_CFG_SDK_PATH = 1,
    NET_EHOME_SS_INIT_CFG_CLOUD_TIME_DIFF = 2,
    NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT = 3,
    NET_EHOME_SS_INIT_CFG_LIBEAY_PATH = 4,
    NET_EHOME_SS_INIT_CFG_SSLEAY_PATH = 5,
    NET_EHOME_SS_INIT_CFG_SQLITE3_PATH = 6
)NET_EHOME_SS_INIT_CFG_TYPE
```

Members

NET_EHOME_SS_INIT_CFG_SDK_PATH

Set the loading path of the storage management module (valid only for Linux operating system)

NET_EHOME_SS_INIT_CFG_CLOUD_TIME_DIFF

Set the time difference of cloud storage request, unit: minute. The value range is 5 to 60, and the default value is 15.

NET_EHOME_SS_INIT_CFG_PUBLIC_IP_PORT

Set the public network address (only valid when the private network and public network are mapped).

NET_EHOME_SS_INIT_CFG_LIBEAY_PATH

Set the path to the OpenSSL encryption library (ie, libeay32.dll or libcrypto.so). The path must be a full path including the library name.

NET_EHOME_SS_INIT_CFG_SSLEAY_PATH

Set the path to the OpenSSL communication library (ie, ssleay32.dll or libssl.so). The path must be a full path including the library name.

NET_EHOME_SS_INIT_CFG_SQLITE3_PATH

Set the path to the sqlite3 library (sqlite3.dll on Windows, libsqlite3.so on Linux).

A.2.26 NET_EHOME_SS_MSG_TYPE

Enumerates the callback data types of the storage server.

Enumeration definition

```
enum _NET_EHOME_SS_MSG_TYPE_{
    NET_EHOME_SS_MSG_TOMCAT = 1,
    NET_EHOME_SS_MSG_KMS_USER_PWD = 2,
    NET_EHOME_SS_MSG_CLOUD_AK = 3 };
```

member

NET_EHOME_SS_MSG_TOMCAT

Callback Tomcat server information, see the structure *NET_EHOME_SS_TOMCAT_MSG* for details.

NET_EHOME_SS_MSG_KMS_USER_PWD

The username and password for the callback Key Management Server (KMS).

NET_EHOME_SS_MSG_CLOUD_AK

Recall the access key of the 5.0 version ISUP device.

A.2.27 NET_EHOME_SS_TYPE

Enumerates storage service types.

Enumeration definition

```
enum NET_EHOME_SS_TYPE{
    NET_EHOME_SS_TYPE_TOMCAT = 1,
    NET_EHOME_SS_TYPE_VRB, = 2
    NET_EHOME_SS_TYPE_KMS, = 3
    NET_EHOME_SS_TYPE_CLOUD = 4
}NET_EHOME_SS_TYPE;
```

member

NET_EHOME_SS_TYPE_TOMCAT

Tomcat

NET_EHOME_SS_TYPE_VRB

VRB

NET_EHOME_SS_TYPE_KMS

KMS

NET_EHOME_SS_TYPE_CLOUD

ISUP5.0 Cloud Storage Protocol

A.2.28 NET_EHOME_STREAM_TYPE

Enumeration of code stream types.

Enumeration definition

```
typedef enum{
    NET_EHOME_MAIN = 1,
    NET_EHOME_SUB = 2,
    NET_EHOME_THIRD = 3,
}NET_EHOME_STREAM_TYPE;
```

member

NET_EHOME_MAIN

Main stream.

NET_EHOME_SUB

Sub-stream.

NET_EHOME_THIRD

The third stream.

A.2.29 NET_EHOME_TALK_ENCODING_TYPE

Voice intercom encoding type.

Enumeration definition

```
typedef enum tagNET_EHOME_TALK_ENCODING_TYPE{
    ENUM_ENCODING_START = 0,
    ENUM_ENCODING_G722_1, = 1
    ENUM_ENCODING_G711_MU, = 2
    ENUM_ENCODING_G711_A, = 3
    ENUM_ENCODING_G723, = 4
    ENUM_ENCODING_MP1L2, = 5
    ENUM_ENCODING_MP2L2, = 6
    ENUM_ENCODING_G726, = 7
    ENUM_ENCODING_AAC, = 8
    ENUM_ENCODING_RAW = 100
}NET_EHOME_TALK_ENCODING_TYPE;
```

member

ENUM_ENCODING_START

Encoding type starts

ENUM_ENCODING_G722_1

G722_1

ENUM_ENCODING_G711_MU

G711_MU

ENUM_ENCODING_G711_A

G711_A

ENUM_ENCODING_G723

G723

ENUM_ENCODING_MP1L2

MP1L2

ENUM_ENCODING_MP2L2

MP2L2

ENUM_ENCODING_G726

G726

ENUM_ENCODING_AAC

AAC

ENUM_ENCODING_RAW

RAW

A.2.30 NET_EHOME_TERMINAL_PLAY_STATE

Enumerates the playback status of the terminal.

Enumeration definition

```
typedef enum{
    NET_EHOME_INVALID_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;
```

member

NET_EHOME_INVALID_STATE

Invalid state.

NET_EHOME_HDMI

HDMI status.

NET_EHOME_VGA

VGA status.

NET_EHOME_SCHEDULE_PLAY

Start playing the schedule.

NET_EHOME_SCHEDULE_STOP

Stop playing the schedule.

NET_EHOME_SCREEN_OFF

Turn off the screen.

A.2.31 NET_EHOME_TERMINAL_TYPE

Enumerates terminal types.

Enumeration definition

```
typedef enum{
    NET_EHOME_INVALID_TERMINAL_TYPE = 0,
    NET_EHOME_ANDROID_STANDARD = 1,
    NET_EHOME_ANDROID_DECODE = 2,
    NET_EHOME_ANDROID_TOUCH = 3,
}NET_EHOME_TERMINAL_TYPE;
```

member

NET_EHOME_INVALID_TERMINAL_TYPE

Invalid type.

NET_EHOME_ANDROID_STANDARD

Android standard device.

NET_EHOME_ANDROID_DECODE

Android decoding device.

NET_EHOME_ANDROID_TOUCH

Android touchscreen devices.

A.2.32 NET_EHOME_THIRD_PARTY_DATA_TYPE

Enumerates the type of data to be sent.

Enumeration definition

```
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;
```

member

NET_EHOME_DATA_INFO_TYPE

Call data.

NET_EHOME_COMMAND_INFO_TYPE

Call control command.

NET_EHOME_CANCEL_SHOT_PIC

Cancel the pop-up image.

NET_EHOME_START_CAPTURE_PIC

Start taking pictures.

NET_EHOME_STOP_CAPTURE_PIC

Stop showing scary images.

A.2.33 NET_EHOME_TIME_ZONE

Enumerates time zones.

Enumeration definition

```
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,};
```

member

NET_EHOME_ZONE_GMT_LOCAL

The local time zone.

NET_EHOME_ZONE_GMT_0

UTC time zone.

NET_EHOME_ZONE_GMT_PLUS_8

UTC+8

A.2.34 NET_EHOME_TRANSMINT_PROTOCOL

Enumerates the transport protocol types.

Enumeration definition

```
typedef enum{
    NET_EHOME_TCP = 1,
    NET_EHOME_UDP = 2,
```

```
NET_EHOME_MCAST = 3,
}NET_EHOME_TRANSMINT_PROTOCOL;
```

member

NET_EHOME_TCP

TCP

NET_EHOME_UDP

UDP

NET_EHOME_MCAST

MCAST

A.2.35 SEARCH_GET_NEXT_STATUS_ENUM

Enumeration state type.

Enumeration definition

```
typedef enum tagSEARCH_GET_NEXT_STATUS_ENUM{
    ENUM_GET_NEXT_STATUS_SUCCESS = 1000,
    ENUM_GET_NETX_STATUS_NO_FILE, = 1001
    ENUM_GET_NETX_STATUS_NEED_WAIT, = 1002
    ENUM_GET_NEXT_STATUS_FINISH, = 1003
    ENUM_GET_NEXT_STATUS_FAILED, = 1004
    ENUM_GET_NEXT_STATUS_NOT_SUPPORT = 1005
}SEARCH_GET_NEXT_STATUS_ENUM
```

member

ENUM_GET_NEXT_STATUS_SUCCESS

After successfully reading a piece of data, you need to call FindNext again to get the next piece of data after processing this data.

ENUM_GET_NETX_STATUS_NO_FILE

No data was found.

ENUM_GET_NETX_STATUS_NEED_WAIT

The data is not ready yet, so you need to wait and continue calling the FindNext function.

ENUM_GET_NEXT_STATUS_FINISH

All data has been retrieved.

ENUM_GET_NEXT_STATUS_FAILED

Abnormal.

ENUM_GET_NEXT_STATUS_NOT_SUPPORT

The device does not support the operation, and the query type is not supported.

A.2.36 SEARCH_TYPE

Enumeration lookup type.

Enumeration definition

```
typedef enum tagSEARCH_TYPE{
    ENUM_SEARCH_TYPE_ERR = -1,
    ENUM_SEARCH_RECORD_FILE = 0,
    ENUM_SEARCH_PICTURE_FILE = 1,
    ENUM_SEARCH_FLOW_INFO = 2,
    ENUM_SEARCH_DEV_LOG = 3,
    ENUM_SEARCH_ALARM_HOST_LOG = 4
}SEARCH_TYPE_ENUM
```

member

ENUM_SEARCH_TYPE_ERR

Error Type

ENUM_SEARCH_RECORD_FILE

Video Files

ENUM_SEARCH_PICTURE_FILE

Image File

ENUM_SEARCH_FLOW_INFO

Stream Information

ENUM_SEARCH_DEV_LOG

Log files

ENUM_SEARCH_ALARM_HOST_LOG

Alarm panel log files

Appendix B. Appendix

B.1 Event Type and Details

Table B-1 Events uploaded via ISAPI

Event Type	eventType value	pAlarmData (event message)
Motion Detection	VMD	<i>MnNcnMn</i> <i>cnM</i>
Video Blocking	shelteralarm <i>MnNcnV</i>	<i>mnmM</i>
Video Loss	videoloss	<i>MnNcnV</i> <i>mM</i>
Alarm input	IO	<i>MnNcnmn</i> <i>mM</i>
Audio anomaly detection	xctin <i>MnNcnxc</i>	<i>n</i>
Out-of-focus detection	defocus	<i>MnNcnc</i> <i>cnmM</i>
Scene change detection	cncnctin <i>MnNcnncnC</i>	<i>no</i>
Face Detection	cctin <i>MnNcnFc</i>	<i>cnmM</i>
Intrusion Detection	flax	<i>MnNcn</i> <i>cn</i>
Cross-border detection	nctin	<i>XML_EventNotificationAlert_linedetect</i> <i>ion</i>
Entering area detection	regionEntrance <i>MnNcnnn</i>	<i>nc</i>

Event Type	eventType value	pAlarmData (event message)
Leaving area detection	rnxtin	<u>XML_EventNotificationAlert_regionExit</u> <u>at</u>
Loitering Detection	loitering <i>MnNcnn</i>	<u><i>cnnM</i></u>
Crowd Detection	group	<u><i>MnNcnG</i></u> <u><i>ncnmM</i></u>
Fast motion detection	rapidMove	<u><i>MnNcnFM</i></u> <u><i>ncnM</i></u>
Parking Detection	parking	<u><i>MnNcnn</i></u> <u><i>cnnM</i></u>
Left-behind detection	nnB <i>MnNcnnn</i>	<u><i>BcnnM</i></u>
Object removal detection	nB	<u><i>MnNcnbcR</i></u> <u><i>mcnM</i></u>
Passive infrared alarm	THE BRIDGE	<u><i>MnNcnRm</i></u> <u><i>M</i></u>
Hybrid Object Detection	mxrctin <i>NnNcnmx</i>	<u><i>cn</i></u>
Image recognition capture	faceCapture	<u><i>NnNcnFcC</i></u> <u><i>M</i></u>
Queue duration detection	personQueueTime <i>NnNcnWn</i>	<u><i>mcnM</i></u>
Area queue number detection	rnQCntin <i>NnNcnQ</i>	<u><i>nmM</i></u>
Face image comparison results	alarmResult	<u><i>NnNcnFcc</i></u> <u><i>CnnM</i></u>

Event Type	eventType value	pAlarmData (event message)
Image modeling results	faceSnapModeling <i>NnNcnFcM</i>	<i>nM</i>
Vehicle Detection (Automatic Number Plate Recognition) ANPR		<i>MnNcnNRM</i> —
Illegal stall detection	cityManagement <i>NnNcn</i>	<i>VNM</i>
Passenger flow statistics report	Cntin <i>MnNcnC</i>	<i>nnnM</i>
Heatmap Report	heatmap	<i>NnNcnM</i> <i>M</i>
Temperature alarm	TMA	<i>MnNcnm</i> <i>mM</i>
Hard disk error alarm	disk error	<i>MnNcn</i> <i>mM</i>
Illegal login alarm	illaccess	<i>MnNcn</i> <i>nmM</i>
Battery status events	bry	<i>NnNcnBy</i> —

B.2 ISUPSDK Error Codes

The following error codes may be returned during the integration process. You can search for the error type and description based on the returned error code to solve the problem.

Common error codes

Error Name	error code	describe
NET_DVR_NOERROR	0	No errors.
NET_DVR_PASSWORD_ERROR	1	The username or password is incorrect.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_DVR_NOENOUGHPRI	2	permission denied.
NET_DVR_NOINIT	3	SDK not initialized.
NET_DVR_CHANNEL_ERROR	4	The channel number is incorrect.
NET_DVR_OVER_MAXLINK	5	No more devices can be added.
NET_DVR_VERSIONNOMATCH	6	SDK and device versions do not match.
NET_DVR_NETWORK_FAIL_CONNECT 7		Failed to connect to the device. The device is offline or the network Disconnect.
NET_DVR_NETWORK_SEND_ERROR 8		Failed to send data.
NET_DVR_NETWORK_RECV_ERROR 9		Failed to receive data from the device.
NET_DVR_NETWORK_RECV_TIMEOUT 10		Timeout in receiving data from the device.
NET_DVR_NETWORK_ERRORDATA	11	data error.
NET_DVR_ORDER_ERROR	12	The calling order is wrong.
NET_DVR_OPERNOPERMIT	13	permission denied.
NET_DVR_COMMANDTIMEOUT	14	The device timed out while executing the command.
NET_DVR_PARAMETER_ERROR	17	Invalid argument.
NET_DVR_NOSUPPORT	23	The device is not supported.
NET_DVR_DVROPRATEFAILED	29	operation failed.
NET_DVR_DIR_ERROR	40	The path is incorrect.
NET_DVR_ALLOC_RESOURCE_ERROR 41		SDK resource allocation error.
NET_DVR_AUDIO_MODE_ERROR	42	Sound card mode does not match.
NET_DVR_NOENOUGH_BUF	43	Insufficient data or image buffer.
NET_DVR_CREATESOCKET_ERROR	44	Failed to create socket.
NET_DVR_SETSOCKET_ERROR	45	Failed to set up socket.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_DVR_MAX_NUM	46	There are no devices to connect to.
NET_DVR_USERNOTEXIST	47	this user does not exist.
NET_DVR_GETLOCALIPANDMACFAIL 53		Failed to obtain local IP or Mac address.
NET_DVR_VOICEMONOPOLIZE	69	The sound card is already in use.
NET_DVR_CREATEDIR_ERROR	71	Failed to create log file directory.
NET_DVR_BINDSOCKET_ERROR	72	Failed to bind socket.
NET_DVR_SOCKETCLOSE_ERROR	73	The socket is dropped.
NET_DVR_USERID_ISUSING	74	Logout failed. The user is currently performing other operations. do.
NET_DVR_SOCKETLISTEN_ERROR	75	Monitoring failed.
NET_DVR_CONVERT_SDK_ERROR	85	Add to dynamic transcoding link library (systemTransform.dll) failed.
NET_DVR_FUNCTION_NOT_ SUPPORT_OS	98	This operating system does not support this feature.
NET_DVR_USE_LOG_SWITCH_FILE	103	Logging is enabled.
NET_DVR_PACKET_TYPE_NOT_ SUPPORT	105	The code stream encapsulation format is invalid.
NET_DVR_IPID_ERROR	106	The IP address is incorrect during network access configuration.
NET_DVR_STREAM_ENCRYPT_ CHECK_FAIL	130	Stream encryption verification failed
NET_DVR_STREAM_STATUS_NOT_ HOT	131	Devices supporting ISUP version 5.0 have not uploaded code Stream initialization information.
NET_DVR_CERTIFICATE_FILE_ERROR 147		Certificate error.
NET_DVR_LOAD_SSL_LIB_ERROR	148	Failed to load the SSL library.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_DVR_LOAD_ANALYZE_DATA_LIB_ERROR	149	Failed to load frame analysis library.
NET_DVR_LOAD_LIBEAY32_DLL_ERROR	156	Failed to load libeay32.dll library.
NET_DVR_LOAD_SSLEAY32_DLL_ERROR	157	Failed to load the ssleay32.dll library.
NET_ERR_LOAD_LIBICONV	158	Failed to load libiconv2.dll library.
NET_ERR_SSL_CONNECT_FAILED	159	SSL connection failed.
NET_ERR_LOAD_ZLIB	161	Failed to load the zlib.dll library.
NET_PREVIEW_ERR_CHANNEL_BUSY 165		This channel is already streaming.
NET_PREVIEW_ERR_CLIENT_BYSY	166	The stream address is repeated.
NET_PREVIEW_ERR_STREAM_UNSUPPORT	167	Unsupported stream type.
NET_PREVIEW_ERR_TRANSPORT_UNSUPPORT	168	Unsupported transport mode.
NET_PREVIEW_ERR_CONNECT_SERVER_FAIL	169	Failed to connect to the preview streaming server.
NET_PREVIEW_ERR_QUERY_WLAN_INFO_FAIL	170	Failed to query the public network exit address of the device.
NET_PREVIEW_ERR_NO_VIDEO_FAIL 171		No video source.
NET_PREVIEW_ERR_SET_ENCODE_PARAM_FAIL	172	Failed to set encoding parameters.
NET_PREVIEW_ERR_SET_PACK_TYPE_FAIL	173	Failed to set the stream branch type.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_PREVIEW_ERR_NOW_IN_PREVIEW_FAIL	174	The device is already fetching streams and no longer supports prefetching streams.
NET_PREVIEW_ERR_NOW_IN_PRESTREAM_FAIL	175	The device is already prefetching the stream and no longer supports prefetching flow.
NET_PREVIEW_ERR_BREAKOFF_PRESTREAM_FAIL	176	The device triggers another flow and disconnects the previous flow. Prefetch stream.
NET_PREVIEW_ERR_P2P_NOT_FOUND	177	The P2P streaming channel does not exist.
NET_PLAYBACK_ERR_TYPE_UNSUPPORTED	3501	The playback type is not supported.
NET_PLAYBACK_ERR_NO_FILE_MATCHED	3502	File not found.
NET_PLAYBACK_ERR_START_TIME_ERROR	3503	Wrong start time.
NET_PLAYBACK_ERR_STOP_TIME_ERROR	3504	Wrong end time.
NET_PLAYBACK_ERR_NO_FILE_FOUND	3505	There is no video recording during this time period.
NET_PLAYBACK_ERR_CONNECT_SERVER_FAIL	3506	Failed to connect to the playback service.
NET_SS_CLIENT_ERR_KMS_TOKEN_FAIL	3601	Failed to obtain token.
NET_SS_CLIENT_ERR_KMS_UPLOAD_FAIL	3602	Failed to upload image to KMS.
NET_SS_CLIENT_ERR_CLOUD_POOLIST_FILE	3603	Failed to obtain the resource pool.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_SS_CLIENT_ERR_CLOUD_ BESTNODE_FAIL	3604	Failed to obtain the optimal node.
NET_SS_CLIENT_ERR_DELETE_PIC_ FAIL	3606	Failed to delete the file.
NET_SS_CLIENT_ERR_PROTO_UNSAFE 3607		Non-secure protocols are not supported in secure mode.
NET_SS_CLIENT_ERR_FILE_INEXISTED 3608		The image to be deleted or downloaded does not exist.
NET_SS_CLIENT_ERR_AUTH_FAILED 3609		Authentication failed.
NET_SS_CLIENT_ERR_UPLOAD_FAIL 3610		Storage component client Cloud\VRB \Tomcat Image upload failed
NET_DVR_ERR_GENERAL_UNKNOW_ ERROR	3701	unknown mistake.
NET_DVR_ERR_GENERAL_PARSE_ FAILED	3702	Message parsing error.
NET_DVR_ERR_GENERAL_SYSTEM_ ERROR	3703	Internal system error (such as a device system call error wrong).
NET_DVR_ERR_GENERAL_ COMMAND_UNKNOW	3704	Illegal command.
NET_DVR_ERR_GENERAL_ COMMAND_NO_LONGER_ SUPPORTED	3705	Obsolete command.
NET_DVR_ERR_GENERAL_ COMMAND_NOT_SUITABLE	3706	Wrong command.
NET_DVR_ERR_GENERAL_ COMMAND_NOT_ALLOW	3707	Unauthorized command.
NET_DVR_ERR_GENERAL_ CHECKSUM_ERROR	3708	The check code is incorrect.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_DVR_ERR_GENERAL_HEADER_INVALID	3709	Illegal message header.
NET_DVR_ERR_GENERAL_LENGTH_INVALID	3710	Illegal message length.
NET_DVR_ERR_GENERAL_PU_BUSY 3711		The device is not responding.
NET_DVR_ERR_GENERAL_OPERATION_FAILED	3712	Wrong opcode.
NET_DVR_ERR_GENERAL_PU_NO_CRYPTO_FOUND	3713	The device or platform did not find the corresponding encryption algorithm.
NET_DVR_ERR_GENERAL_PU_NO_RESOURCE	3715	No resources available.
NET_DVR_ERR_GENERAL_PU_CHANNEL_ERROR	3716	Channel error.
NET_DVR_ERR_GENERAL_SYSTEM_COMMAND_PU_COMMAND_UNSUPPORTED	3717	Unsupported command.
NET_DVR_ERR_GENERAL_SYSTEM_COMMAND_PU_NO_RIGHTS_TO_DO_COMMAND	3718	permission denied.
NET_DVR_ERR_GENERAL_NO_SESSION_FOUND	3719	No session found.
NET_DVR_ERR_GENERAL_PU_NO_VALID_PRELINK	3720	No P2P prelink resources available.
NET_DVR_ERR_GENERAL_PU_NO_INNER_RESOURCE	3721	No direct connection resources are available.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_DVR_ERR_GENERAL_PU_NO_P2P_RESOURCE	3722	No P2P resources available.
NET_DVR_ERR_GENERAL_PU_NO_UESR	3723	The device is not bound to a user.
NET_DVR_ERR_GENERAL_TICKET_EXPIRED	3724	Access credentials have expired.
NET_DVR_ERR_GENERAL_TICKET_INVALID	3725	The access credentials are invalid.
NET_DVR_ERR_GENERAL_NO_P2PSERVER_RESOURCE	3726	No P2P service available.
NET_DVR_ERR_GENERAL_PU_NOT_FOUND	3727	Device not found or device is on the P2P server Offline.
NET_DVR_ERR_GENERAL_SESSION_FREED	3728	The session has been released.
NET_DVR_ERR_RECORD_SEARCH_START_TIME_ERROR	3729	Error in finding the start time of recording.
NET_DVR_ERR_RECORD_SEARCH_STOP_TIME_ERROR	3730	Error in finding the end time of recording.
NET_DVR_ERR_RECORD_SEARCH_FAIL	3731	Failed to find the video.
NET_DVR_ERR_RECORD_NO_RESOURCE	3732	Recording failed. No storage resources are available.
NET_DVR_ERR_CAPTURE_PIC_LOCAL_FAILED	3733	The local image capture of the device failed.
NET_DVR_ERR_CAPTURE_PIC_APPLY_CACHE_FAILED	3734	Image cache request failed.

ISUPSDK (General) Development Guide

Error Name	error code	describe
NET_DVR_ERR_CAPTURE_PIC_PARSE_ PMS_DOMAIN_FAILED	3735	PMS domain name resolution error.
NET_DVR_ERR_CAPTURE_PIC_ CONNECT_PMS_FAILED	3736	PMS connection failed.
NET_DVR_ERR_CAPTURE_PIC_ CREATE_PMS_PACKET_FAILED	3737	Error in creating PMS message.
NET_DVR_ERR_CAPTURE_PIC_SEND_ PMS_FAILED	3738	PMS sending data error.
NET_DVR_ERR_CAPTURE_PIC_RECV_ PMS_FAILED	3739	PMS received data error.
NET_DVR_ERR_CAPTURE_PIC_PARSE_ PMS_RESPONSE_FAILED	3740	PMS response message parsing error.
NET_DVR_ERR_CAPTURE_PIC_GET_ URL_FAILED	3741	Failed to obtain URL.
NET_ERR_DEV_SLEEP	8301	<p>The device is currently in sleep mode</p>  illustrate <p>In the current state, the device cannot preview, play back, Signaling operations in CMS such as voice intercom and configuration response.</p>
NET_ERR_DEV_GOINGTOSLEEP	8302	<p>About to enter sleep mode, estimated 13 seconds</p>  illustrate <p>The device cannot be woken up at this time until it is put into sleep mode.</p> <p>The action is completed and the device officially enters sleep mode.</p> <p>Only then can you wake up.</p>

Error Name	error code	describe
NET_DVR_INIT_CONFIG_ERROR	100000	The initialization configuration interface was not called.
NET_DVR_LOAD_COMPONENT_ERROR	100001	Failed to load component library.

Voice Intercom Error Codes

Error Name	error code	describe
NET_AUDIOINTERCOM_OK	600	No errors.
NET_AUDIOINTECOM_ERR_NOTSUPORT	601	not support.
NET_AUDIOINTECOM_ERR_ALLOC_MEMORY	602	Failed to apply for memory.
NET_AUDIOINTECOM_ERR_PARAMETER	603	The parameter value is incorrect.
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	The calling sequence is incorrect.
NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	Device not found.
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	Device open failed.
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	The context is incorrect.
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV file error.
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	Invalid WAV data type.

Error Name	error code	describe
NET_AUDIOINTECOM_ERR_ ENCODE_FAIL	610	Encoding failed.
NET_AUDIOINTECOM_ERR_ DECODE_FAIL	611	Decoding failed.
NET_AUDIOINTECOM_ERR_ NO_PLAYBACK	612	Audio playback failed.
NET_AUDIOINTECOM_ERR_ DENOISE_FAIL	613	Noise reduction failed.
NET_AUDIOINTECOM_ERR_ UNKNOWN	619	unknown mistake.

B.3 ISAPI Return Codes

The return codes in the ISAPI integration process are classified according to the status codes of the HTTP protocol. The pre-defined status codes include 1-Normal, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid Message Format, 6-Invalid Message Content, 7-Need to Restart the Device, 8-Batch Operation, 9-Other Errors (No Sub-Status Codes Yet). In addition, each status code contains multiple sub-status codes, and there is a one-to-one correspondence between the return code and the sub-status code.

- statusCode=1 (normal) • statusCode=2 (device busy) • statusCode=3 (device error) •
 - statusCode=4 (invalid operation) •
 - statusCode=5 (invalid message format) •
 - statusCode=6 (invalid message content) •
 - statusCode=7 (restart) • statusCode=8 (batch operation)
-

statusCode=1

Substate code (subStateCode)	Error Code	Description (errorMsg)
ok	0x1	The operation is complete.
riskPassword	0x10000002	Risk code.
armProcess	0x10000005	Arming progress.

statusCode=2

Substate code (subStateCode)	Error Code	Description (errorMsg)
noMemory	0x20000001	The device is low on memory.
serviceUnavailable	0x20000002	service is not available.
upgrading	0x20000003	during upgrade.
deviceBusy	0x20000004	The device is busy or the device is not responding.
reConnectIpc	0x20000005	Reconnect to the video server.
transferUpgradePackageFailed	0x20000006	The device upgrade data transfer operation failed.
startUpgradeFailed	0x20000007	The device failed to start the upgrade.
getUpgradeProcessFailed.	0x20000008	Failed to obtain the upgrade status.
crticx	0x2000000B	An authentication certificate already exists.

statusCode=3

Substate code (subStateCode)	Error Code	Description (errorMsg)
deviceError	0x30000001	Hardware error.
badFlash	0x30000002	Operation Flash error.
28181nni	0x30000003	The 28181 configuration was not initialized.
socketConnectError	0x30000005	Socket connection failed.
receiveError	0x30000007	Failed to receive the response message.

Substate code (subStateCode)	Error Code	Description (errorMsg)
deletePictureError	0x3000000A	Failed to delete image.
pictureSizeExceedLimit	0x3000000C	Image size exceeds limit.
clearCacheError	0x3000000D	Clearing cache failed.
updateDatabasError	0x3000000F	Failed to update database.
searchDatabaseError	0x30000010	Database lookup failed.
writeDatabaseError	0x30000011	Database write failed.
deleteDatabaseError	0x30000012	Database element deletion failed.
searchDatabaseElementError 0x30000013		Failed to obtain the number of database elements.
crxctin 0x30000016		Automatic cloud download upgrade package upgrade failed defeat.
Bxctin	0x30001000	Abnormal HBP.
xctin	0x30001001	UDEP abnormality.
ticrcxctin	0x30001002	tic is abnormal.
xctin	0x30001003	abnormal.
Bxctin	0x30001004	Hbase is abnormal.
rxctin	0x30001005	Spark exception.
yrnxctin	0x30001006	Yarn exception.
ccxctin	0x30001007	Cache exception.
rcxctin	0x30001008	The big data server of the card entrance is abnormal.
cxctin	0x30001009	The facial big data server is abnormal.
wxctin	0x3000100A	The Wi-Fi big data server is abnormal.
rcxctin	0x3000100D	The video parameter structure server is abnormal.

ISUPSDK (General) Development Guide

Substate code (subStateCode)	Error Code	Description (errorMsg)
SSDFileSystemIsError	0x30001013	SSD file system error (non-EXT4 file (The system reports an error message when the
ncnCcyFrF 0x30001014		SSD space for personnel frequency detection Insufficient space.
noLinkageResource	0x30001015	Insufficient linkage resources.
noArmingResource	0x30001016	Insufficient deployment resources.
engineAbnormal	0x30002015	Engine abnormality
nnntitin	0x30002016	Engine initialization
algorithmLoadingFailed	0x30002017	Model loading failed
algorithmDownloadFailed	0x30002018	Model download failed
rmcrytinF	0x30002019	Model decryption failed
unboundChannel	0x30002020	Loading a new model requires deleting channel bindings Certainly
nrRtin 0x30002021		Unsupported resolution
unsupportedSteamType	0x30002022	Unsupported stream type
ncncR	0x30002023	Insufficient decoding resources
ncnnrrmnc 0x30002024		Insufficient engine performance (more than the number of Number of channels)
mrrRtin	0x30002025	The image exceeds the resolution (maximum 4096*4096)
improperPicSize	0x30002026	The image is too large (maximum 5M)
URLDownloadFailed	0x30002027	Image download failed (URL download)
unsupportedImageFormat	0x30002028	The image format is not supported (currently only supports jpg)

ISUPSDK (General) Development Guide

Substate code (subStateCode)	Error Code	Description (errorMsg)
unsupportedPollingIntervalTim It is	0x30002029	The polling interval is too low (needs to be greater than 10s)
exceedImagesNumber	0x30002030	Exceed the limit on the number of pictures (each time the platform Number of pictures sent: 1-100 pictures URL, maximum 100)
unsupportedMPID	0x30002031	The MPID sent does not exist on the device. Updates for this MPID are not supported operate
modelPackageNotMatchLabel 0x30002032		The model does not match the description file
modelPackageNotMatchTask 0x30002033		Task and model type do not match
ncnc	0x30002034	Insufficient remaining space. (When the number does not reach The model package size exceeds the remaining Storage space size cannot be added)
engineUnLoadingModelPackag It is	0x30002035	Task delivery failed. The engine is not bound Fixed model. (Unbinding failed, the current The engine has no bound model.)
engineWithModelPackage	0x30002036	Binding failed. The engine has been bound to a module. Type, please unbind first
modelPackageDelete	0x30002037	Binding failed and the model has been deleted.
deleteTaskFailed	0x30002038	Failed to clear task (user closed the task Return on failure)
modelPackageNumberslimited 0x30002039		Failed to add model package. Number of model packages Limit reached.
modelPackageDeleteFailed	0x30002040	Model package deletion failed.
cbrtinm	0x30002051	Calibration timeout.
captureTimeout	0x30006000	Data collection timed out.

Substate code (subStateCode)	Error Code	Description (errorMsg)
lowScore	0x30006001	The quality of the collected data is low.
uploadingFailed	0x30007004	upload failed.

statusCode=4

Sub-status code ↳ subStateCode	Error Code	Description (errorMsg)
notSupport	0x40000001	not support.
lowPrivilege	0x40000002	permission denied.
brtin 0x40000003		Authentication failed.
methodNotAllowed	0x40000004	The operation method is invalid.
notSetHdiskRedund	0x40000005	Unable to set hard disk redundancy attribute.
nvrtin 0x40000006		The operation is invalid.
nctiv	0x40000007	The device is not activated.
ctiv	0x40000008	The device is activated.
crticryx 0x40000009		The certificate already exists.
USBNotExist	0x40000010	The USB device is not connected.
operateFailed	0x4000000F	operation failed.
upgradePackageMore than 2GB	0x40001000	Please upload an upgrade package smaller than 2G.
IDNotExist	0x40001001	ID does not exist.
nrcrtinrr	0x40001002	The interface operation failed.
yncrntinrr 0x40001003		Synchronization failed.
synchronizing	0x40001004	Syncing.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
importError	0x40001005	Import failed.
mrtin	0x40001006	Importing.
ryx	0x40001007	The file already exists.
invalids	0x40001008	Invalid ID.
backupnodeNotAllowe Log	0x40001009	Login is not allowed on the standby node.
xrtinrrr	0x4000100A	Export failed.
buy	0x4000100B	Exporting.
exportEnded	0x4000100C	Export terminated.
exported	0x4000100D	Exported.
IPOccupied	0x4000100E	The IP address is already in use.
IDAlreadyExists	0x4000100F	That ID already exists.
exportItemsExceedLimi t	0x40001010	The number of exports has exceeded the upper limit.
noFiles	0x40001011	file does not exist.
beingExportedByAnoth erUser	0x40001012	Others are exporting.
nRntictin 0x40001013		Authentication is required after the upgrade.
unitAddNotOnline	0x40001015	The added data analysis server is not online.
unitControl	0x40001016	A data analysis server has been added.
analysis unitFull	0x40001017	The data analysis server addition is full.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
unitSearch	0x4000101A	Failed to query the data analysis server list.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoError	0x4000101C	Failed to obtain data analysis server information.
unitGetNodeInfoError 0x4000101D		Failed to obtain node information.
unitGetNetworkInfoError or	0x4000101E	Failed to obtain network information of the data analysis server.
unitSetNetworkInfoError or	0x4000101F	Failed to set the network information of the data analysis server.
setSmartNodeInfoError 0x40001020		Failed to set node information.
setUnitNetworkInfoError or	0x40001021	Failed to set the data analysis server network information.
unitRestartCloseError 0x40001022		Failed to restart or shut down the data analysis server.
virtualIPNotAllowed	0x40001023	Adding virtual IP addresses is not allowed.
unitInstalled	0x40001024	The data analysis server has been installed.
badSubnetMask	0x40001025	The subnet mask is invalid.
uintVersionMismatched	0x40001026	The data analysis server version does not match.
deviceModelMismatched	0x40001027	Add failed. Device model does not match.
unitAddNotSelf	0x40001028	Adding peripherals is not allowed.
noValidUnit	0x40001029	There is no valid data analysis server.
unitNameDuplicate	0x4000102A	The data analytics server name is duplicated.
deleteUnitFirst	0x4000102B	Please delete the data analysis server that has been added to this node first.
getLocalInfoFailed	0x4000102C	Failed to obtain server information.

Sub-status code ȳsubStateCodeȳ	Error Code	Description (errorMsg)
getClientAddedNodeFailed	0x4000102D	Failed to obtain the added node information of the data analysis server.
taskExit	0x4000102E	The task already exists.
taskInitError	0x4000102F	Task initialization failed.
taskSubmitError	0x40001030	Task submission failed.
taskDelError	0x40001031	Task deletion failed.
taskPauseError	0x40001032	Task pause failed.
Cntinrr 0x40001033		Task launch failed.
taskSeverNoCfg	0x40001035	The full-text search server is not configured.
taskPicSeverNoCfg	0x40001036	The image server is not configured.
taskStreamError	0x40001037	The flow information acquisition is abnormal.
taskRecSDK	0x40001038	Historical recording is not supported.
taskCasaError	0x4000103A	Cascading is not supported.
taskVCARuleError	0x4000103B	Smart rule is invalid.
taskNoRun	0x4000103C	Task not executed.
unitLinksNoStorageNone	0x4000103D	The data analysis server is not associated with a node. Please configure the node first.
searchFailed	0x4000103E	Video file search failed.
searchNull	0x4000103F	There is no video footage.
rcffln	0x40001040	The task scheduling service is offline.
updateTypeUnmatched	0x40001041	The upgrade package type does not match.
userExist	0x40001043	User already exists.
userCannotDelAdmin 0x40001044		The admin user cannot be deleted.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
userInexistence	0x40001045	Username does not exist.
userCannotCreatAdmi n	0x40001046	Unable to create admin user.
monitorCamExceed	0x40001048	A maximum of 3,000 monitoring points can be deployed.
monitorCunitOverLimit 0x40001049		Add failed. The control center can be nested up to 5 levels deep.
monitorReginOverLimit 0x4000104A		Add failed. The maximum number of nested regions is 5.
monitorArming	0x4000104B	This monitoring point has been linked to the defense. Please first disarm and then re-arm. try.
monitorSyncCfgNotSet 0x4000104C		Management platform parameters are not configured.
monitorFdSyncing	0x4000104E	Synchronizing. Please wait until the synchronization is complete before continuing.
monitorParseFailed	0x4000104F	Failed to parse monitoring point information.
monitorCreatRootFaile d	0x40001050	Failed to create resource node.
deleteArmingInfo	0x40001051	This monitoring point has been armed. Please touch the arming button before operating.
cannotModify	0x40001052	Modification is not allowed. Please select again.
cannotDel	0x40001053	Deletion is not allowed. Please select again.
deviceExist	0x40001054	The device already exists.
IPErrorConnectFailed 0x40001056		Connection failed. Please check the network port.
cannotAdd	0x40001057	Only supports adding snapshot cameras.
serverExist	0x40001058	The server already exists.
fullTextParamError	0x40001059	The full-text search parameters are incorrect.
largeParamError	0x4000105A	The storage server parameters are incorrect.
picServerFull	0x4000105B	The image server storage space is full.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
NTPUnconnect	0x4000105C	Failed to connect to NTP server. Please check the parameters.
storSerConnectFailed 0x4000105D		Storage server connection failed. Please check the network port.
storSerLoginFailed	0x4000105E	Storage server login failed. Please check your username and password.
searchSerConnectFailed	0x4000105F	Failed to connect to the full-text search server. Please check the network port.
searchSerLoginFailed 0x40001060		Failed to log in to the full-text search server. Please check your username and password.
CnncF	0x40001061	Connection failed. Please check the network port.
mgmtConnectFailed	0x40001062	Failed to connect to the management platform. Please check the network port.
mgmtLoginFailed	0x40001063	Failed to log in to the management platform. Please check your username and password.
TDAConnectFailed	0x40001064	Failed to connect to the traffic data access server. Please check the service Device status.
86sdkConnectFailed	0x40001065	8600 The platform listening port connection failed. Please check the parameters Configuration.
nameExist	0x40001066	A server with the same name exists.
batchProcessFailed	0x40001067	Batch processing failed.
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachesLimit	0x40001069	The service addition limit has been reached.
invalidServiceType.	0x4000106A	Wrong service type.
clusterGetInfo	0x4000106B	Failed to obtain cluster group information.
clusterDelNode	0x4000106C	Failed to delete the node.
clusterAddNode	0x4000106D	Failed to add node.
clusterInstalling	0x4000106E	Cluster assembly in progress...Do not operate.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
clusterUninstall	0x4000106F	Cluster reset...Do not operate.
clusterInstall	0x40001070	Failed to form a cluster.
clusterIpError	0x40001071	The IP address of the task scheduling server is invalid.
clusterNotSameSeg	0x40001072	The active and standby nodes need to be in the same network segment.
clusterVirIpError	0x40001073	Failed to automatically obtain the virtual IP address. Please enter it manually.
clusterNodeUnadd	0x40001074	The specified master (standby) node has not been added.
crNffln 0x40001075		The task scheduling server is offline.
nodeNotCurrentIP	0x40001076	When adding a primary or backup node, you need to select the IP address corresponding to the current page. Analysis node.
addNodeNetFailed	0x40001077	Failed to add node. Network connection failed.
needTwoMgmtNode 0x40001078		When setting up a Zhubei node, two management nodes are required.
Cnc	0x40001079	The virtual IP conflicts with the IP address of the data analysis server.
ipUsed	0x4000107A	The virtual IP address is already in use.
cloudAlalyseOnline	0x4000107B	The cloud analytics server is online.
virIP&mainIPnotSame NetSegment	0x4000107C	The virtual IP address and the active and standby node IP addresses are not on the same network. part.
getNodeDispatchInfoFailed	0x4000107D	Failed to obtain node scheduling information.
unableModifyManage mentNetworkIP	0x4000107E	Failed to modify the management network port. The analysis board is in the cluster.
notSpecifyVirtualIP	0x4000107F	The virtual IP address must be specified for the active and standby clusters.
armingFull	0x40001080	The upper limit of the number of devices to be deployed has been reached.
armingNoFind	0x40001081	Arming information does not exist.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
disArming	0x40001082	Disarm failed.
getArmingError	0x40001084	Failed to obtain the arming information.
refreshArmingError	0x40001085	Failed to refresh the arming information.
ArmingPlateSame	0x40001086	The license plate number of the armed vehicle is the same.
ArmingParseXLSError 0x40001087		Failed to parse the arming information file.
ArmingTimeError	0x40001088	The arming time period is incorrect.
ArmingSearchTimeError	0x40001089	Error retrieving time period.
rmnRtinR cm	0x4000108A	The number of deployment relationships has reached the upper limit.
duplicateArmingName It is	0x4000108B	The deployment relationship name already exists.
noMoreArmingListAdded	0x4000108C	The upper limit of the unauthorized list database has been reached.
noMoreCamerasAdded 0x4000108D		The upper limit of monitoring points has been reached.
noMoreArmingListAddedWithCamera	0x4000108E	The monitoring point associated database has reached the upper limit.
noMoreArmingPeriodAllowed	0x4000108F	The deployment time period has reached the upper limit.
armingPeriodsOverlap	0x40001090	The arming time periods overlap.
noArmingAlarmInfo	0x40001091	Arming alarm information does not exist.
armingAlarmUnRead 0x40001092		Failed to obtain the number of unread arming alarm messages.
getArmingAlarmError 0x40001093		Failed to obtain the arming alarm information.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
searchByPictureTimed Out	0x40001094	The feature search task has timed out. Please search again.
comparisonTimeRange Error	0x40001095	The comparison time range is incorrect.
selectMonitorNumber UpperLimit	0x40001096	The camera ID filter exceeds the upper limit.
noMoreComparisonTas ksAdded	0x40001097	The number of simultaneous comparison tasks exceeds the upper limit.
GetComparisonResultF ailed	0x40001098	Failed to obtain the comparison result.
comparisonTypeError 0x40001099		The comparison type is incorrect.
cmrnnn 0x4000109A		Comparison not completed.
facePictureModelInvali d	0x4000109B	Invalid face model.
duplicateLibraryName. 0x4000109C		Library name already exists.
noRecord	0x4000109D	There is no such record.
cntinRcrF 0x4000109E		Failed to count the number of records.
getHumanFaceFrameF ailed	0x4000109F	Failed to obtain the face frame based on the image.
modelingFailed.	0x400010A0	Modeling based on face image URL failed.
1V1FacePictureCompar isonFailed	0x400010A1	1V1 face similarity comparison failed.
libraryArmed	0x400010A2	The unauthorized list database has been deployed.
licenseExceedLimit	0x400010A3	Dongle limitations.
licenseExpired	0x400010A4	Dongle expired

Sub-status code ȏsubStateCodeȏ	Error Code	Description (errorMsg)
licenseDisabled	0x400010A5	Dongle not available.
licenseNotExist	0x400010A6	The dongle does not exist.
SessionExpired	0x400010A7	The session has expired.
beyondConcurrentLimi t	0x400010A8	Concurrency limit exceeded.
stopSync	0x400010A9	Stop synchronization.
getProgressFaild	0x400010AA	Failed to get progress.
uploadExtraCaps	0x400010AB	The upload file size exceeds the upper limit.
timeRnrrr	0x400010AC	Wrong time range.
dataPortNotConnected	0x400010AD	The data port is not connected.
addClusterNodeFailed	0x400010AE	Failed to add to cluster. The device has already been added to another cluster.
taskNotExist	0x400010AF	The task does not exist.
taskQueryFailed	0x400010B0	Task query failed
modifyTimeRuleFailed	0x400010B2	The task already exists. Modifying the time rule is not allowed.
modifySmartRuleFailed	0x400010B3	The task already exists. Modifying VAC rules is not allowed.
queryHistoryVideoFaile d	0x400010B4	Failed to query historical recordings.
addDeviceFailed	0x400010B5	Failed to add device.
addVideoFailed	0x400010B6	Failed to add the video file.
deleteAllVideoFailed	0x400010B7	Failed to delete all video files.
createVideoIndexFailed	0x400010B8	Failed to create and retrieve the video file.
videoCheckTypeFailed	0x400010B9	The video file type verification failed.
cncrcc F	0x400010BA	Failed to configure the IP address of the fabric server.

Sub-status code ↳ subStateCode	Error Code	Description (errorMsg)
cnrrvr ↳ rF	0x400010BB	Failed to configure the IP address of the image storage server.
storageServiceIPNotExi st	0x400010BD	The storage server IP address does not exist.
syncBackupDatabaseFa iled	0x400010BE	Failed to synchronize the standby database. Please try again.
syncBackupNTPTimeFa iled	0x400010BF	Failed to synchronize the standby server's NTP time.
clusterNotSelectLoopb ackAddress	0x400010C0	Active/standby clusters do not support loopback addresses.
addFaceRecordFailed 0x400010C1		Failed to add face record.
deleteFaceRecordFaile d	0x400010C2	Failed to delete face record.
modifyFaceRecordFaile d	0x400010C3	Face record modification failed.
queryFaceRecordFailed 0x400010C4		Face record query failed.
faceDetectFailed	0x400010C5	Face detection failed.
libraryNotExist	0x400010C6	The library does not exist.
bcQryxrtin 0x400010C7		Exporting matching unauthorized list.
blackListQueryExporte d	0x400010C8	The matching unauthorized list has been exported.
bcQryx rtin	0x400010C9	Stop exporting matching unauthorized lists.
bcrmQryx rtin	0x400010CA	Exporting matching unauthorized list alerts.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
blackListAlarmQueryEx ported	0x400010CB	Matching unauthorized list alerts have been exported.
bcrmQry buy	0x400010CC	Stop exporting matching unauthorized list alerts.
getBigDataCloudAnalysis isFailed	0x400010CD	Failed to obtain big data cloud analysis.
setBigDataCloudAnalysis isFailed	0x400010CE	Failed to configure big data cloud analysis.
submitMapSearchFaile d	0x400010CF	Failed to submit the feature search task.
cnrRtinN x	0x400010D0	The arming relationship does not exist.
getHistoryAlarmInfoFai led	0x400010D1	Failed to obtain historical alarm information.
getFlowReportFailed 0x400010D2		Failed to obtain the passenger flow statistics report.
addGuardFailed	0x400010D3	Failed to add the arming configuration.
deleteGuardFailed	0x400010D4	Deletion of the arming configuration failed.
modifyGuardFailed	0x400010D5	Failed to modify the arming configuration.
queryGuardFailed	0x400010D6	Arming configuration query failed.
uploadUserSuperCaps 0x400010D7		The user information upload limit has been reached.
bigDataServerConnect Failed	0x400010D8	Big data connection failed.
microVideoCloudRequ estInfoBuildFailed	0x400010D9	Weishi Cloud request information establishment failed.
microVideoCloudRespo nseInfoBuildFailed	0x400010DA	Failed to parse the Weishi Cloud response information.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
transcodingServerRequestInfoBuildFailed	0x400010DB	The transcoding server request information failed to be assembled.
transcodingServerResponseInfoParseFailed	0x400010DC	Failed to parse the transcoding server response information.
rncnrvrflin	0x400010DD	The transcoding server is offline.
mcrVCfflin 0x400010DE		Weishi Cloud Service is not online.
rvrfflin	0x400010DF	UPS detection service is not online.
titicRrRq nBF	0x400010E0	Statistics report request information assembly failed.
titicRrRn nrF	0x400010E1	Failed to parse the statistics report response information.
yCnnB F	0x400010E2	Display configuration information failed to be built.
yCnnr F	0x400010E3	Display configuration information parsing failed.
yCnnvF 0x400010E4		Displays configuration information saving failure.
notSupportDisplayConfigType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	wrong password.
upgradePackageLarge 0x400010EB		The upgrade package is too large.
sessionUserReachesLimit	0x400010EC	The upper limit of the number of users logged in through the session has been reached.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
ISO 8601TimeFormatError	0x400010ED	The ISO8601 time format is incorrect.
crtinF 0x400010EE		Cluster disbanding failed.
getServiceNodeInfoFail ed	0x400010EF	Failed to obtain service node information.
getUPSInfoFailed	0x400010F0	Failed to obtain UPS configuration information.
titicRr F	0x400010F1	Failed to obtain the data statistics report.
yCnnF 0x400010F2		Display configuration acquisition failed.
namingAnalysisBoardN otAllowed	0x400010F3	Modification of the analysis board name is not allowed.
onlyDrawRegionsOfCo nvexPolygon	0x400010F4	Only convex polygonal regions are supported.
bigDataServerRespons eInfoParseFailed	0x400010F5	Failed to parse the response information of the big data service.
bigDataServerReturnFa iled	0x400010F6	The big data service returns a failure.
microVideoReturnFaile d	0x400010F7	The Weishi Cloud service returns a failure.
transcodingServerRetu rnFailed	0x400010F8	The transcoding service returned a failure.
UPSServerReturnFailed 0x400010F9		The UPS detection service returned a failure.
forwardingServer ReturnFailed	0x400010FA	The forwarding service returned a failure.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
storageServer ReturnFailed	0x400010FB	The storage service returned a failure.
cloudAnalysisServerRet urnFailed	0x400010FC	The Cloud Analytics service returned a failure.
modelEmpty	0x400010FD	The model obtained is empty.
mainAndBackupNodeC annotModifyManagem entNetworkInterfaceIP	0x400010FE	It is not allowed to modify the management network port IP addresses of the active and standby nodes.
IDTooLong	0x400010FF	ID is too long.
pictureCheckFailed	0x40001100	Image deletion failed.
pictureModelingFailed 0x40001101		Image modeling failed.
setCloudAnalisisDefault ProvinceFailed	0x40001102	Failed to set the default province for the Cloud Analysis Service.
nctinrNmb rxcm	0x40001103	The upper limit of the number of detection areas has been reached.
picturePixelsTooLarge 0x40001105		The image resolution is too high.
picturePixelsTooSmall 0x40001106		The image resolution is too low.
storageServiceIPEmpty 0x40001107		Storage server IP address is empty.
bigDataServerRequestIn foBuildFail	0x40001108	Failed to form the big data service request information.
analysiTimedOut	0x40001109	Analysis timed out.
high- performanceModeDisa bled.	0x4000110A	Please turn on high performance mode.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
cnnrMn rnrvrm	0x4000110B	UPS detected server configuration timeout. Please check the IP address.
cnyRq nrmtinBF	0x4000110C	Failed to construct the cloud analysis service request information.
cnyRn nrmtinrF	0x4000110D	Failed to parse the Cloud Analysis Service response information.
allCloudAnalysisInterfaceFailed	0x4000110E	Failed to call the cloud analysis interface.
cloudAnalysisModelCompareFailed	0x4000110F	Cloud analysis model comparison failed.
cnyFcc rQyRtinF	0x40001110	Cloud analysis face quality scoring failed.
cloudAnalysisExtractFeaturePointsFailed	0x40001111	Cloud analysis feature point extraction failed.
cloudAnalysisExtractPropertyFailed	0x40001112	Cloud analysis property extraction failed.
getAddedNodeInformationFailed	0x40001113	Failed to obtain information about nodes added on the data analysis server.
noMoreAnalysisUnitsAllowed	0x40001114	The upper limit of the number of data analysis servers has been reached.
ctinrnv	0x40001115	The detection area is invalid.
shieldAreaInvalid	0x40001116	The masked region is invalid.
noMoreShieldAreasAdded	0x40001117	The maximum number of blocked areas has been reached.
onlyAreaOfRectangleShapeAllowed	0x40001118	The detection area only supports drawing rectangles.

Sub-status code ȳsubStateCodeȳ	Error Code	Description (errorMsg)
numberReachedLlimit 0x40001119	0x40001119	The limit has been reached.
wait1~3MinutesGetIPAf setupDHCP	0x4000111A	After configuring DHCP, please wait 1 to 3 minutes to obtain an IP address.
plannedTimeMustbeH alfAnHour	0x4000111B	Schedule time must be a full half hour.
oneDeviceCannotBuild Cluster	0x4000111C	A single device cannot form an active/standby cluster.
updatePackageFileNot Uploaded	0x4000111E	The upgrade package file was not uploaded.
rrmnc Nrrwn ctinRn	0x4000111F	High performance mode does not support drawing detection areas.
controlCenterIDDoesN otExist	0x40001120	Control Center ID does not exist.
regionIDDoesNotExist 0x40001121	0x40001121	The region ID does not exist.
licensePlateFormatErro r	0x40001122	The license plate format is incorrect.
mnmnN sNotSupportThisOperat ion	0x40001123	The operation is not supported.
rcBycrRr cNCn	0x40001124	No feature search conditions are configured.
vFnctin FrmNo	0x40001125	The video file packaging format is not supported.
videoPackageFailure 0x40001126	0x40001126	Video transcoding failed.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
videoCodingFormatNot Supported	0x40001127	The video encoding format is not supported.
monitorOfDeviceArmin gdeleteArmingInfo	0x40001129	The monitoring point has been deployed. Please delete the deployment information before operating.
getVideoSourceTypeFai led	0x4000112A	Failed to obtain the video source type.
smartRulesBuildFailed	0x4000112B	Failed to build smart rules.
smartRulesParseFailed	0x4000112C	Failed to parse smart rules.
timRBF	0x4000112D	Formation time rule failed.
timRrF	0x4000112E	Failed to parse time rules.
monitoInfolnvalid	0x4000112F	The monitoring point information is invalid.
addingFailedVersionMi smatches	0x40001130	Add failed. Device version mismatch.
nrmtinRrn ftrCny my	0x40001131	The information returned by the cloud analysis service is empty.
ctinr nrNF Cc	0x40001132	Failed to select the IP address of the master and backup nodes. Please check the status of the master and backup nodes. state.
theSearchIdDoesNotEx is	0x40001133	The search ID does not exist.
incrntin Nx	0x40001134	The sync ID does not exist.
theUserIdDoesNotExist	0x40001136	User ID does not exist.
theIndexCodeDoesNot Exist	0x40001138	The index code does not exist.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
theControlCenterIdDoesNotExist	0x40001139	Control Center ID does not exist.
theRegionIdDoesNotExist	0x4000113A	The region ID does not exist.
theArmingLinkageIdDoesNotExist	0x4000113C	The arming relation ID does not exist.
theListLibraryIdDoesNotExist	0x4000113D	The list library ID does not exist.
invalidCityCode	0x4000113E	Invalid city code.
synchronizingThePasswordOfSpareServerFailed	0x4000113F	The backup system password synchronization failed.
tinrmnyNo.	0x40001140	Modifying the stream type is not supported.
switchingScheduledTaskToTemporaryTaskIsNotSupported	0x40001141	Switching from a scheduled task to a temporary task is not supported.
switchingTemporaryTaskToScheduledTaskIsNotSupported	0x40001142	Switching from ad hoc tasks to scheduled tasks is not supported.
Ncrtin	0x40001143	The task is not scheduled or is being updated.
thisTaskDoesNotExist	0x40001144	The task does not exist in the Cloud Analysis Service.
duplicatedSchedule	0x40001145	The planned time periods cannot overlap.
cntincWmrmymBMr	0x40001146	The planned times with the same algorithm type and consecutively need to be merged.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
invalidStreamingTimeRange	0x40001147	The time range for obtaining traffic is invalid.
invalidListLibraryType	0x40001148	Wrong list library type.
theNumberOfMatchedResultsShouldBeLargerThan0	0x40001149	The number of query results should be greater than 0.
invalidValueRangeOfSimilarity	0x4000114A	The similarity interval is wrong.
nvrtiny	0x4000114B	Invalid sort type.
noMoreListLibraryCanBeLinkedToTheDevice	0x4000114C	The number of lists exceeds the upper limit of a single machine.
InvalidRecipientAddressFormat	0x4000114D	Results The recipient address format is incorrect.
crtinCrFnN	0x4000114E	Insert the dongle before forming a cluster.
theURLIsTooLong	0x4000114F	The task has no configured schedule.
ncCnrFr	0x40001150	The scheduled task has no scheduled time.
theDongleIsExpired	0x40001151	The dongle has expired.
nxctin 0x40001152		The dongle is abnormal.
invalidKey	0x40001153	The authorization service key value is invalid.
crytinF	0x40001154	Authorization service decryption failed.
ncrytinF	0x40001155	Authorization service encryption failed.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
AuthorizeServiceRespo nseError	0x40001156	The authorization service responded abnormally.
incorrectParameter	0x40001157	The authorization service parameters are incorrect.
rtinF	0x40001158	The authorization server operation failed.
noAnalysisResourceOr NoDataInTheListLibrary	0x40001159	Cloud Analysis has no resources or the list library data is empty.
cctinxctin 0x4000115A		The computing node is abnormal.
flax	0x4000115B	Lists being distributed.
rtinN rByCn ytic	0x4000115C	The Cloud Analytics service does not support this operation.
Cnyticn rr	0x4000115D	Cloud Analytics service operations are interrupted.
theServiceIsNotReady 0x4000115E		The service is not ready.
searchingForExternalA piFailed	0x4000115F	Querying the external interface failed.
noOnlineNode	0x40001160	The number of online nodes is empty.
noNodeAllocated	0x40001161	The number of allocated nodes is empty.
noMatchedList	0x40001162	The query list is empty.
ctinFM nyFccr	0x40001163	Allocation failed. Too many big data service lists.
searchIsNotCompleted SearchAgain	0x40001164	The current query is not complete. Please search again.
ctinNCm 0x40001165		The allocation list is not complete.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
rcnFrCn yticRF	0x40001166	The query to the Cloud Analytics service timed out.
noDataOfTheCurrentLi braryFound	0x40001167	The current database has no data. Please make sure there is data in Hbase first.
noFacePictureLibraryls Armed	0x40001168	Currently, there is no defense library for facial big data.
nvbcn Vrnrrnrmtinr mFrnc	0x40001169	The baseline version information is invalid.
crtin cnxctin	0x4000116A	Benchmarking failed. Cannot trigger benchmarking again.
slicinDataFailedNoArm edFacePictureLibrary	0x4000116B	The benchmark failed. There is no corresponding deployment in the face big data information.
GenerateBenchmarkFil eFailedSlicingAgain	0x4000116C	Failed to generate the benchmark file. Slice again.
NonprimaryNodesPro hibitedFromSlicingData	0x4000116D	Non-cluster master nodes cannot perform benchmark operations.
NoReadyNodeToCluste rServers	0x4000116E	Failed to form cluster. No ready node.
NMnmnrv cfln	0x4000116F	The node management service is offline.
theCamera(s)OfTheCo ntrolCenterAreAlready Armed.DisarmThemFir st	0x40001170	There are already monitoring points associated with this control center. Please unlock it first. Try again after removing it.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
theCamera(s)OfTheAreaAreAlreadyArmed.DisarmThemFirst	0x40001171	There are already monitoring points in this area. Please cancel the connection first. Try again.
cnrnfrequencyPeopleDetectionFailed	0x40001172	High frequency detection configuration failed.
searchingForHigh-frequencyPeopleDetectionLogsFailed.	0x40001173	Detection event record query failed.
nrcfrequencyPeopleDetectionLogsFailed.	0x40001174	Failed to detect the details of the query search results.
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	There are already monitoring points associated with this control center.
disarmingFailedTheCamerasNotArmed	0x40001177	Failed to disarm. The monitoring point is not armed.
noDataReturned	0x40001178	The big data service returns an empty result.
preallocFailure	0x40001179	Failed to pre-allocate algorithm resources.
overDogLimit	0x4000117A	Configuration failed. The number of pre-allocated resources has reached the upper limit of the dongle.
analysisServicesDoNotSupport	0x4000117B	The Cloud Analysis Service does not support this feature.
commandAndDispatchServiceError	0x4000117C	Cloud Analytics Scheduling Service Error.
engineModuleError	0x4000117D	Cloud Analytics Engine module error.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
streamingServiceError	0x4000117E	Cloud Analytics stream fetching component error.
faceAnalysisModuleError	0x4000117F	Cloud Analytics face analysis module error.
vehicleAnalysisModuleError	0x40001180	Cloud Analytics vehicle image analysis module error.
videoStructuralAnalysisModuleError	0x40001181	Cloud Analytics video structuring module error.
postprocessingModuleError	0x40001182	Cloud analysis post-processing module error.
rqnyr rnrmryC nrFrbrry	0x40001183	The unauthorized list library has been armed with a high-frequency alarm.
crtinbrryF	0x40001184	Failed to create the list library.
nvntiryy brry	0x40001185	The list library ntiyy is invalid.
noMoreDevicesCanBeArmed	0x40001186	The upper limit of the number of monitoring points has been reached.
nrmyF rvCF	0x40001187	Failed to allocate task resources.
nFrqncy rnctinrmn rmtinF	0x40001188	Failed to set high frequency alarm.
nvrcCntin	0x40001189	The result is invalid.
noMoreTasksCanBeAdded	0x4000118A	The number of tasks has reached the upper limit

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
theTaskIsNotComplete d	0x4000118B	The task is not completed.
resourceOverRemainLi with	0x4000118C	The upper limit of the number of pre-allocated resources has been reached.
frequentlyAppearedPe rsonAlarmsIs ryCnrFr CmrrmFr nry	0x4000118D	The monitoring point has been configured with a high-frequency alarm. Please delete the deployment information first Then operate again.
notEnoughStorageAvail able	0x4000121E	Insufficient available storage space.
wctimfflm 0x4000123B		The time difference between power on and power off needs to be less than 10 minutes.
noData	0x4000124B	Export device has no data.
TasksNumOverRemain Limit	0x40001280	The number of tasks created exceeds the number of tasks allowed to be configured.
noMorePeopleNumCh angeRulesAdded	0x4000128A	The number of occupancy change rules has reached the upper limit.
nMrVnMtin R	0x4000128D	The maximum number of vigorous exercise rules has been reached.
nMrvtinR 0x4000128E		The number of unattended rules for posts has reached the upper limit.
SMRDiskNotSupportRa id	0x40001291	Due to SMR hard drive limitations, RAID is not supported (for performance reasons).
OnlySupportHikAndCus tomProtocol	0x400012a3	Currently only supported via device network SDK and custom protocols Add IPv6 camera.

Substate code (subStateCode)	Error Code	Description (errorMsg)
vehicleEnginesNoReso CER	0x400012a6	The vehicle's engine is running low on resources.
noMoreRunningRulesA well	0x400012a9	The number of personnel running rules has reached the upper limit.
noMoreGroupRulesAd ded	0x400012aa	The upper limit of the number of gathering rules has been reached.
noMoreFailDownRules Added	0x400012ab	The upper limit of the number of man-down rules has been reached.
noMorePlayCellphone RulesAdded	0x400012ac	The upper limit of the number of mobile phone rules has been reached.
rcctin rtinNnn in you m	0x400012B3	The retention time of a fire escape item is greater than or equal to its associated engine polling time.
that day cmNGrr nrcctin rtin	0x400012B4	The polling time of the engine is less than or equal to the fire channel item retention time of its associated channel.
vtinyN bring wcm	0x400012B5	The job dereliction of duty time is greater than or equal to its associated engine polling time.
that day cmNGrr nvtiny	0x400012B6	The polling time of the engine is less than or equal to the unattended time of the position of its associated channel.
cityManagementAlarm IntervalTimeNotLessTh anAnalysisEnginesAuto SwitchTime	0x400012B7	The interval between smart city management alarms is greater than or equal to the polling time of its associated engine.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
analysisEnginesAutoSw itchTimeNotGreaterTh anCityManagementAla rmlIntervalTime	0x400012B8	The polling time of an engine is less than or equal to the intelligence of its associated channel The interval between urban management alarms.
taskBusy	0x400012B9	The task is performing other operations and cannot respond to this operation. (For example, the operation is being paused and you cannot check the progress or restart start)
my 0x400012BA		Path is empty.
nn 0x400012BB		The path is too long.
Nx 0x400012BC		The path does not exist.
noMoreUnregisteredSt reetVendorRulesAdded	0x400012BD	The limit for illegal street vendors has been reached.
noMoreStallOutsideSh opRulesAdded	0x400012BE	The maximum number of off-site operation rules has been reached.
noMoreStallOccupying RoadRulesAdded	0x400012BF	The upper limit of the number of road occupation business rules has been reached.
noMoreIllegalHeapRul esAdded	0x400012C0	The upper limit of the number of random pile rules has been reached.
noMoreIllegalParkingof Non- MotorVehicleRulesAdd ed	0x400012C1	The number of rules for illegal parking of non-motor vehicles has reached the upper limit.
nMrr vrtimnR	0x400012C2	The maximum number of outdoor advertising rules has been reached.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
noMorePackGarbageRulesAdded	0x400012C3	The number of packed garbage rules has reached the upper limit.
noMoreStallUnderUmbrellaRulesAdded	0x400012C4	The number of violations of umbrella-holding rules has reached the upper limit.
nMrbnvrwR	0x400012C5	The maximum number of trash overflow rules has been reached.
noMoreExposeGarbageRulesAdded	0x400012C6	The upper limit of the number of exposed junk rules has been reached.
and MoreHangClothingAlongStreetRulesAdded	0x400012C7	The upper limit for the number of street drying rules has been reached.
ruleEventTypeDuplicate	0x400012C8	Duplicate rule event type.
noCDsInCakeBoxwithEmptyCDs	0x400012CC	There is no disc in the disc bucket.
vcrintin	0x400012CD	The device is printing.
CDWarehouseDoorsOpen	0x400012CE	The disc bin is open.
plateListIsBindingChannel	0x400012D0	The license plate database has been associated with the channel, please disassociate and try again.
cnNmWvrw	0x400012D1	The number of license plates will exceed the upper limit.
rtin	0x400012D2	The license plate data is being processed. The operation is not allowed.
channelNotReady	0x400012D3	The smart channel is not enabled or initialized. Please try again later. Rule configuration.
frequentFormatDisk	0x400012D4	The formatting operation was too frequent. Please try again later.
getFileTimeError	0x400012D5	Failed to obtain the video file time.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
nMrRntinR 0x400015ad		The upper limit of the number of personnel retention rules has been reached.
noMoreSleepOnDutyRulesAdded	0x400015ae	The upper limit of the number of dereliction of duty rules has been reached.
PlateListIDNotExist	0x400015BB	The license plate library ID does not exist.
onlyForHikProtocolAccess	0x400015B	Non-Hikvision protocol access, does not support advanced motion detection.
onlyH264AndH265	0x400015BD	The encoding format is wrong, only H264 and H265 are supported.
vrRtnm 0x400015BE		Resolution exceeded.
PlateIDNotExist	0x400015BF	License plate ID does not exist.
noMoreVMDHumanVehicle	0x400015C0	The upper limit of the number of advanced motion detection channels has been reached.
rryvrw 0x400015C1		The task array exceeds the capacity.
polygonNotAllowCrossing	0x400015C2	Polygons are not allowed to cross.
coverTemplateBinded 0x400015D3		The template has been bound. Please change the template binding.
noEncodedResource 0x400015E8		Insufficient encoding resources available on the device.
Nmn 0x400015EF		The file name length exceeds the limit.
ncnnqFc	0x400015F0	Insufficient storage space for interrogation files.
videoPlayerFileSizeExceedLimit	0x400015F1	The uploaded player size exceeds the limit.
builtInFileName	0x400015F2	The file name is illegal and conflicts with the internal file name of the device.
NotAssociatedWithOwnChannel	0x400019C1	Not associated with own channel.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
AI Target BPCapture Fail 0x400019C5		AI target comparison benchmark image capture failed.
AI Target BPToDSP Fail 0x400019C6		The AI target comparison benchmark image failed to be sent to DSP.
AI Target BP Duplicate Name	0x400019C7	AI target comparison benchmark image names are duplicated.
audioFileNameWrong 0x400019D0		The audio file name is incorrect.
audioFileImportFail	0x400019D1	Failed to import audio file.
audioChannelInProgress	0x400015D4	The audio channel is busy.
recommendAsyncSearch	0x400015F3	The data range is large, please use asynchronous query
trajectoryDataSetInUse 0x400015F6		Personnel movement training data is in use
trajectoryDataSetExistend	0x400015F5	Personnel movement training data already exists
Non-operational hot standby.	0x400019F0	
MaximumNumberOfDevices	0x400019F1	The number of devices has reached the upper limit.
StandbyMachineCannotBeDeleted	0x400019F2	This hot standby machine cannot be deleted.
userCertificateNotInstalled	0x40002036	The user certificate is not installed.
CA Certificate Not Installed	0x40002037	The CA certificate is not installed.
Failed to Verify Certificate	0x40002038	Failed to verify certificate. Please check the certificate.
userCertificateExpired	0x40002039	The user certificate has expired.
clientCertificateRevocation	0x4000203A	The user certificate has been revoked.

Sub-status code ↳ subStateCode	Error Code	Description (errorMsg)
CCrticxr 0x4000203B		The CA certificate has expired.
CCrticRvcti n	0x4000203C	The CA certificate has been revoked.
connectFail	0x4000203D	Connection failed.
loginNumExceedLimit 0x4000203F		The number of logged-in users has reached the upper limit.
rmnF	0x40002056	Failed to format the hard drive. Please try again.
ncryFrmnF 0x40002057		Failed to format the encrypted hard drive.
wrongPassword	0x40002058	Failed to verify the SD card password. Wrong password.
audiolsPlayingPleaseW belonging to	0x40002067	The report is being broadcast. Please wait.
twoWayAudioInProgre ssPleaseWait	0x40002068	Voice intercom in progress, please wait.
cbrtinnNmF 0x40002069		The number of calibration points has reached the upper limit.
cmvCb rtinFr	0x4000206A	Horizontal calibration not set.
cmRrCm rCbrtinFr	0x4000206B	Radar dome calibration is not set.
HBDLibNumOverLimit 0x40002092		The human body bank has reached its limit
theShieldRegionError 0x40002093		Setting failed. Need to draw the ground area where the shielded object is located area.
ctinrrrr 0x40002094		Setup failed. Need to draw the ground area where the detection range is located.
invalidLaneLine	0x40002096	Saving failed. Lane markings do not meet requirements.
pointsOnStraightLine 0x4000209C		Calibration failed. The calibration points cannot be on a straight line.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
TValueLessThanOrEqua lZero	0x4000209D	Calibration failed. The T value of the calibration point must be greater than 0.
NotSupportWithVideo Task	0x400020F3	The channel has enabled video structuring and does not support this function.
nctinr	0x400050df	Configuration failed. No detection zone was set.
storSerDownloadFileFa iled	0x40006000	Failed to download the file from the storage service
storSerDownloadFileSi zeZero	0x40006001	The size of the file downloaded from the storage service is 0
illegalTaskID	0x40006002	The task ID is invalid.
rrNCn	0x40006003	Storage service not configured
armingFailed	0x40008000	Arming failed.
disarmingFailed	0x40008001	Disarm failed.
clearAlarmFailed	0x40008002	Failed to clear the alarm information.
bypassFailed	0x40008003	Bypass failed.
bypassRecoverFailed	0x40008004	Bypass recovery failed.
outputsOpenFailed	0x40008005	Failed to open the relay.
outputsCloseFailed	0x40008006	Failed to close the relay.
registerTimeOut	0x40008007	Registration timed out.
registerFailed	0x40008008	registration failed.
addedByOtherHost	0x40008009	The peripheral has already been added by another alarm control panel.
alreadyAdded	0x4000800A	Peripherals added.
armedStatus	0x4000800B	The system is armed.
bypassStatus	0x4000800C	Set to bypass state.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
zoneNotSupport	0x4000800D	This operation is not supported within the zone.
zoneFault	0x4000800E	The zone is in the preset state.
wCnc	0x4000800F	Password conflict.
audioTestEntryFailed 0x40008010		Failed to start audio test mode.
audioTestRecoveryFailed	0x40008011	Failed to turn off audio test pattern.
addCardMode	0x40008012	Add card model.
searchMode	0x40008013	Query mode.
addRemoterMode	0x40008014	Added remote control mode.
registerMode	0x40008015	Registration mode.
exDevNotExist	0x40008016	The peripheral does not exist.
theNumberOfExDevLimited	0x40008017	No addable peripherals.
rnCnF	0x40008018	Siren setup failed.
chanCannotRepeatedBinded	0x40008019	The channel has been associated with the zone.
inProgramMode	0x4000801B	The keyboard is in programming mode.
inPaceTest	0x4000801C	Walk test mode.
arming	0x4000801D	Armed.
masterSlavesEnable 0x4000802C		The slave radar does not support this operation. The master-slave relationship is in effect.
forceTrackNotEnabled 0x4000802D		Forced follow disabled.
NrnCn Bycr	0x4000802E	The current area does not support setting this zone type.
alarmLineCross	0x4000802F	Alarm wires crossed.

ISUPSDK (General) Development Guide

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
zoneDrawingOutOfRange	0x40008030	The drawn zone is outside the detection range.
alarmLineDrawingOutOfRange	0x40008031	The drawn alarm line exceeds the detection range.
hasTargetInWarningArea	0x40008032	There is a target in the warning area. Do you want to enable forced arming?
radarModuleConnectFailed	0x40008033	Radar module communication failed.
importCfgFilePasswordErr	0x40008034	The password was incorrect when importing the configuration file.
overAudioFileNumLimit	0x40008038	The number of audio files exceeds the limit.
audioFormatIsWrong	0x4000803a	The audio file format is incorrect.
audioFileIsLarge	0x4000803b	The audio file is too large.
beyondNumberLimit	0x4000a012	Exceeded the limit
subscribeTypeErr	0x4000a016	Subscriptions are not supported for this metadata type.
EISError	0x4000A01C	When the intelligent function is turned on, the electronic image stabilization fails.
jpegPicWithAppendDataError	0x4000A01D	Failed to capture the thermal image. The temperature measurement configuration parameters are abnormal (reflectivity, distance). If the distance or reflected temperature is too large or too small, it will cause the heat map to be captured fail).
MRtin	0x40002040	HDMI video resolution cannot be larger than the main stream and sub stream resolution.
startAppFail	0x400020B2	Failed to launch the application.
yvcnc	0x400020B3	The original video stream conflicts.
overMaxAppNum	0x400020B4	The maximum number of application uploads has been reached.

ISUPSDK (General) Development Guide

Sub-status code ȏsubStateCodeȏ	Error Code	Description (errorMsg)
alreadyExist	0x40002029	The application already exists.
noFlash	0x400020B5	Not enough Flash.
alreadyRunning	0x40002026	The application is running.
notRunning	0x40002027	The application has stopped.
packNotFound	0x40002028	The package does not exist.
noMemory	0x4000202A	Not enough storage.
hdFormatFail	0x40002049	Failed to format the hard drive.
raidLicense	0x4000202B	The license is invalid.
CnmCrmC n	0x400020C4	The operation failed. Please confirm whether the classroom is in interaction.
rftFrCncW flax	0x40004006	Forklift filtering and ship detection are mutually exclusive. Please turn off ship detection. Retry.
ctinCnc WFrftFr	0x40004007	The forklift filter under the ship detection and temperature measurement functions are mutually exclusive. Please turn off Disable forklift filtering and try again.
rftFrCncW WGctin	0x40004008	Forklift filtration and exhaust gas detection are mutually exclusive. Please turn off exhaust gas detection. Retry.
wasteGasDetectionConf lictsWithForkliftFilter	0x40004009	The forklift filter under the exhaust gas detection and temperature measurement functions are mutually exclusive, please turn off Disable forklift filtering and try again.
rftFrCncW Vcctin	0x4000400A	Forklift filtering and vehicle detection are mutually exclusive. Please turn off vehicle detection. Retry.
vcctinCn cWFrftFr	0x4000400B	The forklift filter under the vehicle detection and temperature measurement functions are mutually exclusive. Please turn off Disable forklift filtering and try again.
nRctinBrF WFnCbrti n	0x4000400C	If the sun reflection filter fails to be turned on, please contact a professional Dual-light fusion calibration.

Sub-status code ÿsubStateCodeÿ	Error Code	Description (errorMsg)
emptyEventName	0x400015E0	Event description cannot be empty.
sameEventName	0x400015E1	The same event description exists.
emptyEventType	0x400015E2	Event type cannot be empty.
sameEventType	0x400015E3	The same event type exists.
maxEventNameReached	0x400015E4	Exceeded the maximum number of event descriptions supported.
hotSpareNotAllowedExternalStorage	0x400015FC	The external storage cannot be opened in hot standby mode.
sameCustomProtocolName	0x400015FD	The same protocol name exists.
maxPTZTriggerChannelReached	0x400015FE	The upper limit of PTZ linkage channels has been reached.
POSCanotAddHolidayPlan	0x400015FF	Holiday scheduling does not support POS events.
eventTypeIsTooLong 0x40001600		Event type is too long.
eventNameIsTooLong 0x40001601		Event name is too long.
PerimeterEnginesNoResource	0x40001602	Insufficient perimeter engine resources.
invalidProvinceCode 0x40001607		Illegal province code.
trainAndScoreNotExecutedTogether	0x4000160A	Only one of the training task and the scoring task is allowed to be executed at the same time.

statusCode=5

Substate code (subStateCode)	Error Code	Description (errorMsg)
badXmlFormat	0x50000001	The XML is malformed.

statusCode=6

Substate code (subStateCode)	Error Code	Description (errorMsg)
badParameters	0x60000001	Parameter error.
badHostAddress	0x60000002	The host IP address is incorrect.
badXmlContent	0x60000003	XML content error.
badIPv4Address	0x60000004	The IPv4 address is incorrect.
badIPv6Address	0x60000005	The IPv6 address is incorrect.
cncv4r	0x60000006	IPv4 address conflict.
cncv6r	0x60000007	IPv6 address conflict.
badDomainName	0x60000008	Wrong domain name.
connectSreverFail	0x60000009	Connection failure.
cncmnNm	0x6000000A	Domain name conflict.
badPort	0x6000000B	Port number conflict.
portError	0x6000000C	Port error.
exportErrorData	0x6000000D	Importing data failed.
badNetMask	0x6000000E	The subnet mask is incorrect.
badVersion	0x6000000F	Version mismatch.
badDevType	0x600000010	Device type mismatch.
badLanguage	0x600000011	Language mismatch.
incorrectUserNameOrPasswor d	0x600000012	incorrect username or password.
invalidStoragePoolOfCloudServ is	0x600000013	The cloud storage pool is invalid. The storage pool or storage pool ID is incorrect.
noFreeSpaceOfStoragePool	0x600000014	Insufficient storage pool space.
riskPassword	0x600000015	The password is not secure.

Substate code (subStateCode)	Error Code	Description (errorMsg)
UnSupportCapture	0x600000016	After Smart264 is turned on, it is not possible to capture Resolution is 4096*2160 or 3072*2048 picture.
userPwdLenUnder8	0x60000023	The password must contain at least two of the following: Numbers, uppercase and lowercase letters, special characters symbol.
userPwdNameSame	0x60000025	The username and password are the same.
userPwdNameMirror	0x60000026	The password cannot be the username written backwards.
beyondARGSRangeLimit	0x60000027	The parameter value exceeds the limit.
ctinnctinR n	0x60000085	The rule line is not within the area.
ctinRnrrr	0x60000086	The rule area is incorrect. Please make sure the rule The region is a convex polygon.
ctinRnCntin Rn	0x60000087	The rule area must be marked with a red frame.
PedalAreaError	0x60000088	The pedal area must be placed within the regulation area.
ctinrBrrr	0x60000089	Detection zones A and B must be within the rule box Inside.
ABRegionCannotIntersect	0x6000008A	Areas A and B cannot overlap.
customHBPIDError	0x6000008b	Incorrect custom body library image ID.
customHBPIDRepeat	0x6000008c	Duplicate custom body image ID.
dataVersionsInHBDLibMismatc hes	0x6000008d	The database version in the human body library does not match.
invalidHBPID	0x6000008e	The human data PID is invalid.
invalidHBDID	0x6000008f	The body bank ID is invalid.

ISUPSDK (General) Development Guide

Substate code (subStateCode)	Error Code	Description (errorMsg)
humanLibraryError	0x60000090	The human body database data is incorrect.
humanLibraryNumError	0x60000091	Exceeds the maximum number of human bodies in the database.
humanImagesNumError	0x60000092	Exceeds the maximum number of human body images.
noHumanInThePicture	0x60000093	Modeling failed, no human body was identified.
analysisEnginesNoResourceErr or	0x60001000	No analysis engine resources.
analysisEnginesUsageExced	0x60001001	Engine usage is exceeding capacity.
PicAnalysisNoResourceError	0x60001002	No corresponding intelligent analysis engine supports graph Secondary identification of the film.
analysisEnginesLoadingError	0x60001003	The analysis engine is initializing.
analysisEnginesAbnormalError 0x60001004		Analysis engine exception.
nynnFcbrtin 0x60001005		The image is being imported into the face database. Edit Failed to parse engine parameters.
analysisEnginesAssociatedChan In the	0x60001006	The analytics engine is associated with the channel.
smdEncodingNoResource	0x60001007	Insufficient motion detection encoding resources.
smdDecodingNoResource	0x60001008	Insufficient motion detection decoding resources.
diskError	0x60001009	Hard disk error.
diskFull	0x6000100A	The hard disk is full.
facelibDataProcessing	0x6000100B	Processing face database data.
capturePackageFailed	0x6000100C	Packet capture failed.
capturePackageProcessing	0x6000100D	Capturing packets.
noSupportWithPlaybackAbstra ct	0x6000100E	This feature is not supported. Video is enabled Summary playback.
ncnNwrBnw 0x6000100F		Insufficient network bandwidth.

ISUPSDK (General) Development Guide

Substate code (subStateCode)	Error Code	Description (errorMsg)
tapeLibNeedStopArchive	0x60001010	Please stop library archiving first.
ntiyyrrr	0x60001011	The interactive password is incorrect.
ntiyyMn	0x60001012	The interactive password is missing.
noSupportWithPersonDensityDetect	0x60001013	This feature is not supported. Density detection.
cRtinvrw	0x60001014	IPC resolution exceeded.
cBrvrw	0x60001015	The IPC bit rate exceeds the limit.
Grmrnc 0x60001016		The time difference between the device and the server is too large big.
noSupportWithPlayback	0x60001017	This feature is not supported. Playback is enabled.
channelNoSupportWithSMD	0x60001018	This feature is not supported. Mobile is enabled Detection.
channelNoSupportWithFD	0x60001019	This feature is not supported. Image is enabled Identify and capture.
illegalPhoneNumber	0x6000101A	Invalid phone number.
DashNmbr	0x6000101B	The document number is invalid.
linkedCameraOutLimit	0x6000101C	Connection to monitoring point timed out.
achieveMaxChannelLimit	0x6000101E	The maximum number of channels has been reached.
humanMisInfoFilterEnabledChanNumError	0x6000101F	The number of human body false alarm channels opened reached To the upper limit.
humanEnginesNoResource	0x60001020	The People Analytics Engine is running low on resources.
Nmbrrvrw	0x60001021	The maximum number of tasks has been reached.
cnmvrw 0x60001022		The collision time limit has been reached.
invalidTaskID	0x60001023	The task ID is invalid.
eventNotSupport	0x60001024	The event subscription is not supported.

Substate code (subStateCode)	Error Code	Description (errorMsg)
invalidEZVIZSecretKey	0x60001034	The Hik-Connect authentication code is invalid.
nbVrctin 0x60001042		Two-factor authentication required.
nbVrctinr 0x60001043		No two-factor authentication users.
timnNmrvrm	0x60001044	The maximum number of time periods has been reached.
channelNumOverLimit	0x60001045	The maximum number of channels has been reached.
noSearchIDResource	0x60001046	Insufficient query ID resources.
oneKeyExport	0x60001047	The export operation is in progress, please try again later try.
oneKeyRemove	0x6000104e	Deletion is in progress, please try again later try.
noSupportDeleteStrangerLib 0x60001051		Stranger list detection is not supported.
noSupportCreateStrangerLib 0x60001052		Creation of stranger list library is not supported.
behaviorAnalysisRuleInfoError 0x60001053		The abnormal behavior detection rule parameters are incorrect.
safetyHelmetParamError	0x60001054	Wrong helmet parameters.
OneChannelOnlyCanBindOneEngine i have	0x60001077	A channel can only be bound to one engine. Now an engine is bound. Cannot bind again.
mtinrrr	0x6000107B	The image location coordinates are out of range.
xtinrrr	0x6000107C	The text position coordinates are out of range.
LogDiskNotSetReadOnlyInGroupMode	0x60001100	The log disk in the disk group cannot be set Read-only.
LogDiskNotSetReDundancyInGroupMode	0x60001101	The log disk in the disk group cannot be set For backup.
AudioFileNameDuplicate	0x60001135	The audio file name is duplicated.

Substate code (subStateCode)	Error Code	Description (errorMsg)
CurrentAudioFileAIRuleInUseAI readyDelete	0x60001136	The AI rules used by the current audio file The linkage has been deleted.
rntinmmc	0x60002000	Failed to open the device. EMMC use over.
tivrmNnb 0x60002001		The bitrate adaptation function is not enabled.
tivrmnVrbB rnb	0x60002002	Bitrate adaptation and dynamic bitrate cannot Enable both.
noSafetyHelmetRegion	0x60002023	The helmet detection area is not configured.
unclosedSafetyHelmet	0x60002024	Hard hat detection is enabled.
width/ Rticrrrr	0x6000202C	The maximum aspect ratio of the uploaded image is 2:1, minimum is 1:2.
Nnti	0x6000202E	The gimbal is not initialized.
PTZMoving	0x6000202F	The PTZ is in motion.
PTZLocked	0x60002030	The gimbal is locked.
deployExceedMax	0x60006018	The maximum number of arming connections has been reached.
detectorTypeMismatch	0x60008000	Detector type mismatch.
nameExist	0x60008001	Name already exists.
uploadImageSizeError	0x60008016	The size of the uploaded image cannot exceed 5MB
laneAndRegionOverlap	0x60008013	Channels overlap.
nCrtinNnc /		The unit parameter is invalid.
rnnMCnc /		The regular area overlaps with the shielded area.
wholeRuleInShieldingMask	/	The temperature measurement rules for the shielded area have been configured.
holidayNameContainChineseOr SpecialChar	0x60001080	Holiday names do not support Chinese and special character.

Substate code (subStateCode)	Error Code	Description (errorMsg)
genderValueError	0x60001081	The gender value is invalid.
certifyVrrr 0x60001082		The certificate type is invalid.
personInfoExtendValueIsTooLong	0x60001083	The length of the custom label in the face database exceeds the limit.
personInfoExtendValueContainsIllegalChar	0x60001084	The custom label of the face database contains illegal character.
excelHeaderError	0x60001085	Error in Excel header.

statusCode=7

Substate code (subStateCode)	Error Code	Description (errorMsg)
rebootRequired	0x70000001	Restart to take effect.

statusCode=8

Substate code (subStateCode)	Error Code	Description (errorMsg)
notAllSuccess	0x80000001	The bulk operation was partially successful.
allSuccess	0x80000002	All batch operations were successful.
allFailed	0x80000003	All batch operations failed.

B.4 Functional Modular Error Code

The functional modular error codes returned during the ISAPI integration process are classified according to the product functional modules. Error code, error code

Explanations and troubleshooting suggestions are provided in the table below.

Common function module (error code range: 0x00000000, 0x00100001 to 0x001)

Error code string error code		Error code description	Troubleshooting suggestions
success	0x00000000 Success		
vcNctiv 0x00100001	The device is not activated.		Activate the device.
deviceNoPermission	0x00100002 Device operation failed due to permissions not enough.		Update the permissions of the operation user.
deviceNotSupport	0x00100003 The device does not support this feature.		Distinguish the device capability level and call the corresponding supported functional interfaces.
deviceResourceNotEnough	0x00100004 The device is running out of resources.		Release device operating resources.
dataFormatError	0x00100005 The message format does not conform to the syntax (JSON) Require.		
resetError	0x00100006 Restore to factory default after power failure. Newly activated, due to reset key status The device is abnormal, usually stuck.		
parameterError	0x00100007 Parameter error.		
	0x00100100 The channel is illegal.		Please confirm whether the channel is legal.
	0x00100101 The encrypted code stream does not support NPQ. Please use other methods to obtain encrypted code streams. Stream mode.		
	0x00100102 The number of NPQ channels has reached the upper limit. Please reduce the number of NPQ channels and try again. Try to get the flow.		
	0x00100103 Unsupported stream type.		Please check the class of the stream request Type.
	0x00100104 The number of connections exceeds the limit.		Please reduce the number of streaming clients and try again. New flow.
	0x00100105 Insufficient bandwidth.		Please reduce the number of remote flow lines.

User function module (error code range: 0x00200001 to **0x002**)

Error code string error code		Error code description	Troubleshooting suggestions
passwordError	0x00200001	The username and password are incorrect.	Confirm the correct password.
userNameNotExist	0x00200002	The user name does not exist.	Verify that the username exists, or Re-add the username.
userNameLocked	0x00200003	The user name is locked.	Wait for the device to unlock.
userNumLimited	0x00200004	The upper limit of user login.	Log out.
lowPrivilege	0x00200005	Insufficient permissions for this operation do.	<p>For user operations, take the following measures</p> <p>Troubleshooting:</p> <ul style="list-style-type: none"> • Cannot delete itself • Cannot modify own user level or Permissions • User permissions are low and cannot obtain high Permission user information • Users cannot upgrade to low-privilege users Level and authority <p>For other types of operations, follow the following measures</p> <p>Troubleshooting: Unable to configure user permissions</p> <p>External operation, check your own use</p> <p>User type and permissions; if you are still unsure</p> <p>If you agree, please contact R&D after obtaining the information.</p>
incorrectUserNameOrPassword	0x00200006	The username or password is incorrect. Check that the configured	<p>username and password are correct.</p> <p>No, if you are not sure, please contact</p> <p>Administrator reconfigures username or password</p> <p>If the administrator password is forgotten,</p> <p>You can reset the password of the device</p>
riskPassword	0x00200007	Risky password	The password complexity is too low, please modify it again password.

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
passwordMustContainMorethan8Characters	0x00200008	The password length must be greater than or equal to 8. Check whether the configured password length is greater than or equal to 8. If not, modify the password again.	
passwordLenNoMoreThan16	0x00200009	The password length cannot be greater than 16. Check whether the configured password length is greater than 16. If you want to change your password again	
adminUserNotAllowedModify	0x0020000a	Unable to modify admin information. Check whether the modified user is admin. Admin user name, permissions and other information cannot be modified.	
cnrmwrr	0x0020000b	When modifying user information, confirm the password. Error.	
passwordMustContainMorethan2Types	0x0020000c	The password must contain any two or more of numbers, lowercase letters, uppercase letters, and special characters. Check whether the password configuration meets the requirements (the password must contain any two or more of numbers/lowercase/uppercase/special characters).	
passwordContainUserName	0x0020000d	Illegal password. The password cannot contain the username. Check whether the password contains the username.	
userPwdNameMirror	0x0020000e	The password cannot be the username written in reverse. Check that the password contains the username written in reverse.	

Time function module (error code range: 0x00300001 to **0x003**)

Error code string error code		Error code description	Troubleshooting suggestions
failed. manualAdjustmentFailed	0x00300001	The device time calibration	
NTPError	0x00300002	The NTP service address is invalid. Confirm the valid NTP service address.	
timFrmrrr	0x00300003	The time format sent during the time calibration is wrong. The message format is wrong or the time content is wrong. For example, the ISO 8601 time format is wrong. The local time is: 2018-02-	

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
		01T19:54:04, the actual output is: 2018-02-01 19:54:04	
beyondTimeRangeLimit	0x00300004	The time calibration time is not within the range of the time and device capability time supported by the device. If the device supports it, compare to see if it exceeds the range. The time calibration range is: 2000-01-01T00:00:00~2037-12-31T23:59:59, actually issued 2038-01-01T00:00:00	
ntimrrnBntim	0x00300005	The start time of the validity period cannot be later than the end time. Check and compare the start time and end time to see if the validity period is legal (such as the personnel validity period, the start time must be less than the end time, etc.).	

Network function module (error code range: 0x00400001 to **0x004**)

Error code string error code		Error code description	Troubleshooting suggestions
domainNameParseFailed	0x00400001	Domain name resolution failed.	
PPPOEConnectedFailed	0x00400002	PPOE networking failed.	
FTPConnectedFailed	0x00400003	The FTP server is offline.	
vcCnc	0x00400004	IP address conflict.	
libraryConnectedFailed	0x00400005	The image and video library is offline and not connected.	
F	0x00400006	File (including image) upload failed. Check the network connection. If there is no problem, contact after-sales service.	

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
storSerDownloadFileFailed	0x00400007	1. Download files from storage services (including images) failed. 2. File (including image) download failed defeat.	Check the network connection. If there is no problem, Contact after-sales.
storSerDownloadFileSizeZero	0x00400008	Download files from storage service (package The size of the image is 0.)	Check the network connection. If there is no problem, Contact after-sales.
rrNCn	0x00400009	The storage service is not configured.	Check whether the configuration is correct.
badHostAddress	0x0040000a	The host address is incorrect.	Check whether the configuration is correct.
badIPv4Address	0x0040000b	The IPv4 address is incorrect.	Check whether the configuration is correct.
badIPv6Address	0x0040000c	The IPv6 address is incorrect.	Check whether the configuration is correct.
cncv4r	0x0040000d	IPv4 address conflict.	Check the IPv4 address configuration in the network Condition.
cncv6r	0x0040000e	IPv6 address conflict.	Check the IPv6 address configuration in the network Condition.
badDomainName	0x0040000f	Bad domain name	Check whether the configuration is correct.
connectSreverFail	0x00400010	Failed to connect to the server.	Check whether the network is unobstructed and the Check whether the setting is correct.
cncmnNm	0x00400011	Domain name conflict	Check whether the configuration is correct.
badPort	0x00400012	Port conflict	Check whether the configuration is correct.
portError	0x00400013	Port Error	Check whether the configuration is correct.
badNetMask	0x00400014	Subnet mask error	Check whether the configuration is correct.
badVersion	0x00400015	Version mismatch	Check that the version is correct.
badDns	0x00400016	Wrong DNS	Check whether the configuration is correct.
badMTU	0x00400017	Wrong MTU	Check whether the configuration is correct.

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
badGateway	0x00400018	Bad Gateway	Check whether the configuration is correct.
urlDownloadFail	0x00400019	The URL download failed.	Check the network and URL to see if they are correct.
deployExceedMax	0x0040001a	The deployment exceeds the maximum number of connections. Confirm that the maximum supported deployment number has been reached.	The number of defenses deployed.

Maintenance function module (error code range: 0x00500001 to **0x005**)

Error code string error code		Error code description	Troubleshooting suggestions
upgradeXMLFormatError	0x00500001	The upgrade request XML format is incorrect. Check whether the upgrade file is correct. If the class	The upgrade file is correct. Use local upgrade.
upgradeContentError	0x00500002	The upgrade request content is incorrect. Check whether the upgrade file is correct. If the class	The upgrade file is correct. Use local upgrade.
noUpgradePermission	0x00500003	The corresponding user does not have permission to upgrade. Try changing to admin user. Or ask admin to configure advanced operations Authorization	
upgrading	0x00500004	Upgrade in progress	Already upgrading, waiting for the last update completed
receiveUpgradePackageError	0x00500005	Error in receiving upgrade package	Please check whether the network is abnormal.
upgradePackageLanguageMismatch	0x00500006	The upgrade package language does not match	Please confirm the upgrade package language type, Language type
upgradePackageMismatch	0x00500007	The upgrade package does not match the device type. Please confirm the upgrade package device type and device type.	Equipment Type
OEMCodeMismatch	0x00500008	The customized customer's OEM code does not match The upgrade package is incorrect.	Please contact after-sales service to obtain the correct upgrade Level Pack

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
versionMismatch 0x00500009	The upgraded version does not match		Please contact after-sales service to obtain the correct supported upgrade package version
upgradeHalfFailed 0x0050000c	The upgrade process is halfway through when the device prompts that the upgrade has failed (the reason is or cache problem).		
deviceParameterImportFailed	0x0050000d Device parameter import failed (device type inconsistent numbers, versions, platforms, etc.).		
vcnccrytin rrr	0x0050000e When upgrading the corresponding model version, the program prompts that the upgrade package does not match (the reason is that the device has a problem with the equipment encryption, usually after-sales encryption error).		
SDCardFormatError	0x00500025 SD card formatting failed.		
SDCardLoadFailed	0x00500026 The SD card is inserted, but the page is not displayed. Show.		
NASFailed	0x00500027 NAS mount failed.		
hardDiskError	0x00500028 Hard disk abnormality (hard disk does not exist, incompatible, encrypted disk, capacity not supported, format abnormality, array abnormality incompatibility, etc.)		
upgradeError	0x00500030 Upgrade error		From 0x00500030-0x0050005f, allocated for upgrade use.
upgradePackageSizeMismatch	0x00500032 The size of the downloaded upgrade package does not match the size in the upgrade command. The size is obtained when querying the upgrade package information.		
upgradePackageSizeExceeded	0x00500033 The file size exceeds the partition size.		

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
domainNameParseFailedForDownload	0x00500034	The download address domain name resolution failed.	
netWorkUnstable	0x00500035	The network is unstable and the download timed out. Maximum number of retries exceeded).	
digestValueMismatch	0x00500036	The digest value does not match.	
signatureVerifyFailed	0x00500037	Signature verification failed.	
innerFormatError	0x00500038	The upgrade package is damaged (internal format is incorrect)	
memoryNotEnough	0x00500039	Insufficient memory	
burnFailed	0x0050003a	Error in burning firmware	
unknownError	0x0050003b	An unknown error occurred in the underlying interface	
userCancel	0x0050003c	User initiated stop.	
systemResume	0x0050003d	Upgrade failed. Minimal system recovery.	
vctfrmMc	0x00500060	The upgrade package does not match the device platform type. Please confirm the upgrade package platform type. (For example, the upgrade package of the GD platform obtains the device platform type, and then upgrades it to the ST platform device.) For the upgrade package upgrade type, different types are not allowed to upgrade	
upgradeObjectNoSupport	0x00500061	The upgrade object is not supported.	Check the upgrade object and contact R&D to confirm whether the upgrade object supports the upgrade.
upgradeObjectNoOnline	0x00500062	The upgrade target is not online.	Check whether the object to be upgraded is online (check the network for network upgrade and check the network and serial port for network to serial port conversion). Use the tool to obtain the online status. If it is not online,

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
			If you cannot obtain the upgrade package, upgrade Cannot succeed
vrctinF	0x00500063	The upgrade package signature verification failed. Please contact after-sales service for correct support.	If you use the correct upgrade The packet still reports the error, capture the packet in parallel Department of R&D
upgradeDataErrLength	0x00500064	The upgrade package data length is abnormal. Check whether the network is abnormal.	Is the level package normal?
upgradeParameterWrong	0x00500065	Upgrade package parameter error	Including CRC check failure, upgrade package The head is not found. Please contact after-sales service. Get the correct and supported upgrade package version
passwordDecodeError	0x00500066	Password decryption failed.	Password decryption failed, contact R&D, Provide corresponding printing and packet capture
soon	0x00500067	Not initialized	Check whether the configuration file is correct. Is the resource file set to the device correct? Yes, there are still problems after the inspection. Contact R&D and provide corresponding resources document
taskNotExist	0x00500068	The task does not exist.	When templates are sent in batches, the task does not exist. Check whether the operation is correct. Capture packets to obtain corresponding logs and check Check whether the captured packet data meets the protocol requirements Request and contact R&D
noMoreTasksCanBeAdded	0x00500069	The number of tasks has reached the upper limit.	The batch distribution of templates has not been completed. The distribution operation needs to be performed. Get the corresponding log of the package and check the captured package Does the data flow meet the SDK protocol? and contact R&D

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
cnrtin 0x0050006a	The device is being imported or exported.		Allow to operate again
Frmrrr 0x0050006b	The file format is wrong (for example, it supports Excel but Word is sent)		The file format is wrong. Check whether the format of the file being operated is compliant.
Cnnrrr 0x0050006c	File content error (table item or content mistake)		The file content is wrong, similar to table items or internal errors. Check the file content to see if there are any errors.
importEmptyFile 0x0050006d	The file content is empty		The file content is empty. Check whether the file being operated is an empty file.
xcrrr 0x0050006e	The file is too large, including the upgrade package. (for example, 200KB for face delivery and other operations; after eliminating the reasons, if the device has a limit on the image size and has not been located yet, contact R&D again. When the face image is too large, it may return a failure; when manually capturing, the image is too large).		Possible reasons: The face image is too large (for example, 200KB for face delivery and other operations; after eliminating the reasons, if the device has a limit on the image size and has not been located yet, contact R&D again. When the face image is too large, it may return a failure; when manually capturing, the image is too large).
roomNoExisted	0x0050006f	Room number conflict (already exists) Room number conflict (already exists), get the existing room number, and check whether the configured room number exists in the existing room number.	
rNmbrCn c	0x00500070	Serial number conflict	Check the configured parameters to see if the same serial number exists.
vcxCnc 0x00500071	Device number conflict		Check the configured device numbers to see if they are identical.
roomNoStorageFul l	0x00500072	Number storage space is full	Confirm the device capacity. If it is full, storage is not allowed. Clear or delete it before operating.

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
roomNoNotExist 0x00500073		The room number does not exist.	Get and check whether the room number exists. If it does not exist, add it first.
iotNumReachesLi with	0x00500074	The number of IOT channels has reached the upper limit (for example, the upper limit supported by the device has been reached when adding an indoor unit or connecting an external device such as a door station)	
rebootRequired	0x00500075	A reboot is required before the operation takes effect. Before the operation takes effect, a manual reboot is required, otherwise the operation will be invalid.	
updateDatabaseErr or	0x00500076	Database update failed	If the database update fails, contact R&D, provide relevant logs, and check the device.
searchDatabaseErr or	0x00500077	Database search failed	If the database update fails, contact R&D, provide relevant logs, and check the device.
writeDatabaseErro r	0x00500078	Database write failed	If the database update fails, contact R&D, provide relevant logs, and check the device.
deleteDatabaseErr or	0x00500079	Database element deletion failed. Database element deletion failed. Contact R&D, provide relevant logs, and check on the device.	
searchDatabaseEle mentError	0x0050007a	Failed to find the number of database elements. Failed to find the number of database elements. Contact R&D, provide relevant logs, and check on the device.	
	0x00500080	The upgrade file was not found.	Please check whether the upgrade package path is too long or whether there is a correct upgrade package in the upgrade package path.
	0x00500081	The upgrade package engine type does not match. Please select an upgrade package that matches the device engine type.	

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
	0x00500082	Camera domain name resolution failed	<p>Please confirm that the device is configured correctly.</p> <p>Is there a DNS service or camera domain name effect.</p>
	0x00500083	The camera network is unreachable	<p>Please confirm whether the local network is accessible.</p> <p>Add the network where the channel is located.</p>

Preview function module (error code range: 0x00600001 to **0x006**)

Error code string error code		Error code description	Troubleshooting suggestions
liveViewFailed	0x00600001	Preview failed. The number of streaming links reached to the maximum limit.	
	0x00600002	Request encapsulation format is abnormal	<p>Please check the package format for request preview Mode.</p>
	0x00600003	EHome2.x is not supported after it is enabled NPQ	<p>eHome2.x is enabled, please change to He previews the way.</p>
	0x00600005	Zero channel does not support NPQ preview. Please change to other previews for zero channel preview. Way.	
	0x00600007	Only virtual stream supports NPQ. Please change the stream type to virtual stream. flow.	
	0x0060000A	The IP channel is offline	<p>Please confirm that the IP channel is online before trying again.</p> <p>Try previewing.</p>
	0x0060000B	The device does not support transcoding preview. Please use other stream types for preview.	
	0x0060000C	Channel zero is not enabled	<p>To preview the zero channel, please turn on the zero channel first.</p> <p>start.</p>
	0x0060000D	Transcoding capacity exceeds limit	<p>Please reduce the camera resolution or Number of transcoding channels.</p>
	0x00600010	This channel has no sub-stream	<p>This channel is not connected to the sub-stream, please use Preview in main stream mode.</p>

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
	0x00600011	The device does not support NPQ preview. Please change to another preview mode.	
	0x00600012	NPQ function is not enabled	Please enable NPQ function or change other preview methods first.

Playback function module (error code range: 0x00700001 to **0x007**)

Error code string error code		Error code description	Troubleshooting suggestions
	0x00700001	Playback failed because most front-ends currently only support 1-channel playback.	
	0x00700002	The playback magnification on the video wall cannot be reduced.	
	0x00700003	The playback stream push rate is too high. Reduce the push stream rate.	
	0x00700004	The device does not support the playback stream encoding format. Provide the encoding format stream supported by the device.	
	0x00700005	The device does not support the playback stream encapsulation type. Provide the stream encapsulation type supported by the device.	
	0x00700007	Playback decoding abnormality (wall display problem; image/display/decoding abnormality; image freeze/black screen) (unsupported stream type, preview freeze, decoding abnormality abnormal audio decoding, screen distortion)	
	0x00700008	The playback video does not exist or the search failed. Please try to search again or confirm whether the hard disk is normal.	
	0x00700009	Playback time parameter error	Please make sure that the search recording time period is correct before trying to play it back.
	0x0070000A	Invalid recording type	Please select the correct video type to search.

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
	0x0070000B	Invalid time type	Please select the correct time type search.
	0x0070000C	Invalid event parameters	Please select the correct event parameters search.
	0x0070000D	Invalid event type	Please select the correct event type search.
	0x0070000E	The device does not support intelligent retrieval. Please select non-intelligent retrieval method.	search.
	0x0070000F	Invalid smart event type. Please select a correct smart event type.	to search.
	0x00700010	Invalid dynamic analysis sensitivity. Please select the correct sensitivity to record.	Like search.
	0x00700011	Reverse playback is not supported	Please select the correct playback mode.
	0x00700012	Invalid file status	Please select the correct file status search.
	0x00700013	Invalid search starting position. Please use a correct search starting position.	to search.
	0x00700014	Invalid maximum search quantity. Please use a correct maximum search quantity.	Retrieve.

Snapshot function module (error code range: 0x00800001 to **0x008**)

Error code string error code		Error code description	Troubleshooting suggestions
	0x00800001	Manual capture failed.	

Voice intercom module (error code range: 0x00900001 to **0x09**)

Error code string error code		Error code description	Troubleshooting suggestions
startFailed	0x00900001	Failed to start intercom.	
codingFormatNot Match	0x00900002	The encoding formats of the two parties are different Negotiation failed.	View or capture protocol interactions on the platform Message, analyze the audio of the negotiation between the two parties Whether the encoding format is consistent.
dialedIsBusy	0x00900003	The party being intercomed is already in the intercom. No longer able to respond to the intercom.	Check whether the other party is actually talking If it is not the case that the packet capture protocol interaction Message, analyze the other party's response message.
tintinn mbrrrr	0x00900004	The destination long number of the request is incorrect. Check the platform or capture the protocol interaction	Check the platform or capture the protocol interaction Message, analysis trombone.

Video storage module (error code range: 0x00a00001 to **0x00**)

Error code string error code		Error code description	Troubleshooting suggestions
videoSearchFailed	0x00a00001	Video search failed.	The device has no storage resources
notFindStorageMedium	0x00a00002	Storage medium not found.	
videoDownloadFailed	0x00a00003	Video download failed.	
checkMemoryCard	0x40002126	Search failed, please check the storage status Is it normal?	

Image storage module (error code range: 0x00b00001 to **0x0b**)

Error code string error code		Error code description	Troubleshooting suggestions
	0x00b00001	Image query failed	The image has no storage resources

IO function module (error code range: 0x00c00001 to **0x00c)**

Error code string error code		Error code description	Troubleshooting suggestions
	0x00c00001	Invalid alarm input number	
	0x00c00002	Invalid alarm output number	

Event function module (error code range: 0x00d00001 to **0x00)**

Error code string error code		Error code description	Troubleshooting suggestions
	0x00d00001	Event rule configuration error	Refer to the operating specification configuration

Parking service module (error code range: 0x00e00001 to **0x00)**

Error code string error code		Error code description	Troubleshooting suggestions
	0x00e00001	The package car already exists	The car is created based on the license plate. To check if the license plate is used A chartered vehicle has been created.
	0x00e00002	The license plate number parameter should not be a null character string	

General function module (error code range: 0x00f00001 to **0x00)**

Error code string error code		Error code description	Troubleshooting suggestions
noMemory	0x00f00001	Insufficient device memory (heap space allocation fail)	Linux, free in the root directory Memory, free column memory remaining value, The log is sent to R&D for positioning.
deviceBusy	0x00f00002	The device is busy or unresponsive. The log is sent back to R&D for analysis.	Fingerprint collection/face collection/file download Send/upload files related business, confirm Whether the last operation has been completed become

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
notSupport	0x00f00003 Device not supported: URL not found (URL not supported)		1. Use wireshark to capture packets and check the URL packets sent. Check the PMP protocol platform to see if the URL can be searched. If the URL can be searched, send it to R&D for analysis and location.
methodNotAllowed	0x00f00004	HTTP method not allowed: method and protocol do not match, e.g. PUT method protocol, POST is sent	Use wireshark to capture packets, check the URL message sent, and check the method corresponding to the URL by comparing it with the PMP protocol platform.
nvrtin 0x00f00005	Illegal operation of the interface input instruction (for example: the channel number that the device does not support, unsupported hardware, etc. will return an invalid operation when sent)		When the device is implemented, it is required to return the specific error reason in errorMsg
IDNotExist	0x00f00006	ID does not exist: In the URL, use Wireshark to capture packets according to the protocol and check whether the URL should carry an ID, but does not actually carry an ID. Check whether the ID carried is correct.	
invalids	0x00f00007 Invalid ID: When the URL carries an ID, the ID is exceeded. Use Wireshark to capture packets and check whether the URL carries the correct ID if the capability set or data format is not standardized.		If there is a capability set corresponding to the URL, obtain the capability set corresponding to the URL and compare the ID range.
invalidURL	0x00f00008	The content after the "?" question mark in the URL is wrong. URL is correct.	Use wireshark to capture the packet and check whether the URL is correct.
deviceAckTimeOut 0x00f00009	Device response timeout		1. If the communication with the external module times out, first check whether the external module is offline; 2. After eliminating problem 1, contact the equipment R&D department for location after obtaining other logs.

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
badXmlFormat	0x00f0000a	XML format error (for example, when the device is not fully implemented, the errorMsg is required to be complete and cannot be used as a completed node name. Check the XML parsing, such as <a>, which should be based on the errorMsg and the corresponding node <a>)	
badJsonFormat	0x00f0000b	JSON format error: similar to Json format, double quotes missing: "employeeName"XXX"	When the device is implemented, it is required that the errorMsg returns the specific node name. When troubleshooting, it is based on the errorMsg and the corresponding node capabilities.
bathURLFormat	0x00f0000c	URL format error: The URL format is incorrect. Use wireshark to capture the packet and check whether the URL is correct.	
badXmlContent	0x00f0000d	XML message error: <ul style="list-style-type: none">The message only has the URL, no message bodyThe required node exists, no parametersThe parameter range limit is exceeded: the parameter content of the node is wrong	When the device is implemented, it is required to return the specific node name in errorMsg. When troubleshooting, capture the packet to obtain the message, and perform the check based on errorMsg and the corresponding node capabilities.
badJsonContent	0x00f0000e	JSON message error: <ul style="list-style-type: none">The message only has the URL, no message bodyThe required node exists, no parametersThe parameter range limit is exceeded: the parameter content of the node is wrong	When the device is implemented, it is required to return the specific node name in errorMsg. When troubleshooting, capture the packet to obtain the message, and perform the check based on errorMsg and the corresponding node capabilities.
is required to exist. The sLack	0x00f0000f	Message parameter is missing. The required node is not implemented. The errorMsg messageParameter specific node name is returned. When troubleshooting, it is based on errorMsg and the corresponding node capabilities.	

Error code string error code		Error code description	Troubleshooting suggestions
invalidSearchCondit	0x00f00010	The search condition is invalid. Please search again. For search protocol, see ions	Is the searchID correct?
operObjectNotExis	0x00f00011	The operation object does not exist (for doors and alarms, the device needs to be connected to the door lock and card reader. t output and alarm input related operations. It is returned when the object has not been added.)	Check whether the door lock has received the alarm enter.

Door control module (error code range: 0x01000001 to **0x010**)

Error code string error code		Error code description	Troubleshooting suggestions
mtinticti	0x01000001	Multiple authentication state operation failed. When the door is in n Failed	multiple authentication state and the remote door opening step is not executed in the multiple authentication, the remote door opening will return this error.
door cannot be opened.	0x01000002	The security door control module is offline and the n defeat.	Check whether the security door control is offline.

Plan template module (error code range: 0x01100001 to **0x011**)

Error code string error code		Error code description	Troubleshooting suggestions
nNmbrCn	0x01100001	Plan number conflict (for example, there are duplicate holidays in one holiday group, holiday group, plan template number day number, and there is an illegal duplicate holiday group number in one plan template, etc.). You can check the platform message to locate it.	
timvr	0x01100002	Time period conflict (for example, time period overlap conflicts in weekly schedules and holiday schedules)	Check the messages to see if there is any overlap at different times of the day.

Personnel information module (error code range: 0x01200001 to **0x012**)

Error code string error code		Error code description	Troubleshooting suggestions
deviceUserFull	0x01200001	User storage is full (when permissions are issued, the number of personnel reaches the maximum storage capacity of the device, check whether the maximum capacity has been reached.)	Query the number of personnel available on the device. If the number of personnel reaches the maximum storage capacity of the device, check whether the maximum capacity has been reached.
employeeNoNotExis	0x01200002	User does not exist (for personnel management, business such as personnel modification and query parameters exists by work number, if the personnel does not exist, an error code may be returned)	Get personnel by work number, check whether the personnel exists by work number, if the personnel does not exist, an error code may be returned)
employeeNoAlready	0x01200003	The user already exists (in personnel management, yExist adding personnel by work number, if the person already exists, an error code may be returned)	Get personnel by work number, and when adding personnel by work number, if the person already exists, an error code may be returned)

Credential module (error code range: 0x01300001 to **0x013**)

Error code string error code		Error code description	Troubleshooting suggestions
deviceCardFull	0x01300001	Card is full: When the card is issued, the card number queries the number of cards already in the device to see if the maximum storage capacity of the device has been reached.	
cardNoAlreadyExis	0x01300002	Card already exists: When managing the card, press the card number to obtain the card parameters, and check whether the card number exists when adding the card number. If the card number already exists, an error code may be returned.	
cardNoNotExist	0x01300003	Card does not exist: When managing the card, press the card number to obtain the card parameters to see if the card number modification, query parameters and other business numbers exist. If the card does not exist, an error code may be returned.	
cardFullPerUser	0x01300004	Single card is full: When the device obtains the number of cards issued by the current person for each person, associated by the device has been limited, if when adding a new card to a person, it may exceed the maximum capability set; times	whether the number of cards allowed to be

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
moduleHandleFailed	0x01300005	Module command processing failed	R&D investigation
moduleConnectFailed	0x01300006	Module connection failed	Whether the communication cable of the fingerprint module has poor contact;
nrrnFr MrF	0x01300007	Module or algorithm template merging failed (The call may occur when collecting and entering born).	Whether the number of fingerprints collected meets the requirement; Whether the finger is centered when fingerprint collection is performed; Whether the fingerprint is damaged; Whether the collected finger is the same one.
moduleIllegalParam	0x01300008	The module uses incorrect parameters. Please check whether the called instruction parameters are legal.	
nrrnFr Nmnv	0x01300009	The number of module or algorithm fingerprint merging is not available. Whether the number of fingerprints collected has reached the required efficiency (it may be called during collection and entry, usually 3 times).	Whether the number of fingerprints collected has reached the required efficiency (it may be called during collection and entry, usually 3 times).
nrrnNx	0x0130000a	The fingerprint does not exist (check whether the relevant work number/card number has been downloaded from the card reader or the main machine when obtaining the fingerprint by the card number or work number, and whether it has been deleted).	
notSupportDelFPB	0x0130000b	The card reader or host does not support the finger ID delete fingerprint. Device communication protocol version (communication protocol version can be checked through version).	card reader and host version.
nrrnwQ and	0x0130000c	Fingerprint quality is poor (it may occur when the finger is centered during fingerprint collection).	Whether the fingerprint is damaged.
nrrnF	0x0130000d	The fingerprint is full. The fingerprint on the non-module side obtains the card reader parameters and checks the number of fingerprints currently stored;	Capacity Full
nrrnry x	0x0130000e	The fingerprint already exists and is repeated (referring to checking the repeated fingerprint association information to see (may occur when hair is under the tattoo)).	whether the repeated fingerprints are of the same person;

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
			If the person is not the same person, contact algorithm development to check the template matching situation.
no. no yx	0x0130000f	The host already has the fingerprint ID. Is the finger (Finger number 1-10) When adding fingerprints	D sent with the ID the same? Is the fingerprint deleted successfully?
errorFingerPrintTy on	0x01300010	This type of fingerprint template is not supported. Check whether the data logger and the module match.	
nrrnFr r	0x01300011	The number of fingerprints per person has reached the upper limit. Check whether the current person's fingerprints have been fully downloaded; whether the fingerprints are deleted successfully.	
nrrnNn myNrCr N	0x01300012	The fingerprint is not associated with the work ID or card number (check the Number or Card Number field is empty)	capture data to see if the fingerprint is not associated with the work ID or card number. If so, it has been located. Otherwise, provide the capture data to contact R&D.
pupilDistanceTooS mall	0x01300013	The distance between the face and eyes in the image is insufficient: May occur when hair is down the face	If the inter-eye distance returned by DSP is smaller than the inter-eye distance configured by the application, you can obtain the face image and contact R&D for positioning.
faceDetectFailed	0x01300014	Face detection failed: The algorithm engine returns failure when the face is sent. You can contact R&D for positioning after obtaining the face image.	
pictureFaceDetect Zero	0x01300015	There is no face in the picture: the algorithm engine returns success, but the number of faces detected is 0; you can first check whether there is a face in the picture. If there is, you can obtain the face picture and contact R&D for positioning. If there is no face, you can directly reply.	May occur when hair is down the face
can contact R&D after led	0x01300016	Picture decoding failed: The algorithm engine returned a failure when the face was sent. You can contact R&D after obtaining the face picture.	May occur

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
picFormatError	0x01300017	The device does not support the image format: The face DSP bottom layer returns "image format error". It may happen when sending images in bmp, jpg, and png formats.	The face DSP bottom layer returns "image format error". Currently, DSP only supports images in bmp, jpg, and png formats. After eliminating this reason, you can obtain the face image and contact R&D.
pictureModelingFailed	0x01300018	Image modeling failed: The algorithm engine returns a failure when the face is sent. To locate the problem, obtain the face image and contact R&D.	Image modeling failed: The algorithm engine returns a failure when the face is sent. To locate the problem, obtain the face image and contact R&D.
faceLowQuality	0x01300019	Poor face quality: Face collection is usually caused by incorrect posture, too close or too far distance, etc. You can adjust and try again. If it fails continuously, you can shoot the collection video and contact R&D.	Poor face quality: Face collection is usually caused by incorrect posture, too close or too far distance, etc. You can adjust and try again. If it fails continuously, you can shoot the collection video and contact R&D.
faceTypeError	0x0130001a	Error in sending face: Possible reasons for sending face: When the device is in normal mode, it may send infrared face; after eliminating the reasons, if it is still not located, contact R&D	Error in sending face: Possible reasons for sending face: When the device is in normal mode, it may send infrared face; after eliminating the reasons, if it is still not located, contact R&D
deviceFaceFull	0x0130001b	The number of faces is full: When the faces are sent, the total number of faces currently on the device can be obtained to see whether the maximum capacity has been reached. If so, the problem has been located; otherwise, contact R&D.	The number of faces is full: When the faces are sent, the total number of faces currently on the device can be obtained to see whether the maximum capacity has been reached. If so, the problem has been located; otherwise, contact R&D.
deviceUserAlreadyExistFace	0x0130001c	A face already exists for the employee: The face is obtained based on the work number. When the person is issued, it may be returned to see whether the employee's face exists when a new face is added. If so, the problem has been located, otherwise contact R&D.	A face already exists for the employee: The face is obtained based on the work number. When the person is issued, it may be returned to see whether the employee's face exists when a new face is added. If so, the problem has been located, otherwise contact R&D.

Error code string error code		Error code description	Troubleshooting suggestions
cRtInR tirr	0x0130001d	The face resolution or aspect ratio is abnormal: This may happen when the face is sent.	If the algorithm engine returns a failure, you can obtain the face image and contact R&D for positioning.
faceDataNotExist	0x0130001e	User face data does not exist: General platform searches for the face of the person, it may be deleted after issuance. The data can be located by cooperating with the device attachment information event (if the device supports it).	It can be basically determined that the facial data of the person has not been issued on site. When the be possible to check whether the data has been issued or

Safety function module (error code range: 0x01400001 to **0x014**)

Error code string error code		Error code description	Troubleshooting suggestions
decryptFailed	0x01400001	When decrypting sensitive information fields, the keys for importing and exporting decryption must be consistent. Failed; or when importing data files, decryption failed.	
nvCritic	0x01400002	When importing a certificate, the one imported is not a legitimate certificate file; when importing a private key, a public third-party website can be used to verify the legitimacy of the certificate before use.	
crticNmc	0x01400003	The certificates do not match. The public and private certificates uploaded to the device for SSL/TLS need to match in pairs. Both certificates need to be generated at the same time.	
nctiv	0x01400004	The device is not activated.	Before using the device functions, activate the device using tools such as SADP.
ctiv	0x01400005	The device is already activated.	The device has been activated and does not need to be activated again
forbiddenIP	0x01400006	The IP address is prohibited.	If the user logs in illegally for too long, IP lock is prohibited

ISUPSDK (General) Development Guide

Error code string error code		Error code description	Troubleshooting suggestions
bondMacAddressNotMatch	0x01400007	The MAC address does not match the user. The specified user must use a specific MAC address to log in. If a similar error is reported, check whether the user's parameters are bound to a specific MAC.	The MAC address does not match the user. The specified user must use a specific MAC address to log in. If a similar error is reported, check whether the user's parameters are bound to a specific MAC.
bondIpAddressNotMatch	0x01400008	The IP address does not match the user. Specify the user to use a specific IP address To log in. If a similar error is reported, check whether the user's parameters are bound to a specific IP.	The IP address does not match the user. Specify the user to use a specific IP address To log in. If a similar error is reported, check whether the user's parameters are bound to a specific IP.
brtin	0x01400009 Triggered by illegal login.		The device password is incorrect, triggering illegal login

Advertising function module (error code range: 0x01500001 to **0x015**)

Error code string error code		Error code description	Troubleshooting suggestions
materialDownloadFailed	0x01500001	Material download failed. Is the network connection normal? Whether the device is operating normally. See what is printed wrong.	Material download failed. Is the network connection normal? Whether the device is operating normally. See what is printed wrong.
materialNumberIsOver	0x01500002	The program material format exceeds the limit (for example, check whether the number of materials in the program table contains a maximum of 5 materials, and it will be triggered when the sent materials exceed the limit). table sent exceeds the specified limit. A	The program material format exceeds the limit (for example, check whether the number of materials in the program table contains a maximum of 5 materials, and it will be triggered when the sent materials exceed the limit). table sent exceeds the specified limit. A

B.5 Device Log Type

Alarm log

value	describe
0x1	Alarm input
0x2	Alarm Output
0x3	Motion detection alarm starts
0x4	Motion detection alarm ends
0x5	Video blocking alarm starts
0x6	Video blocking alarm ends
0x7	Smart alarm starts
0x8	Smart alarm ends
0x09	Traffic incident alarm starts
0x0a	Traffic incident alarm ended
0x0b	Network alarm starts
0x0c	Network alarm ends
0x0d	Network alarm recovery
0x0e	Wireless alarm starts
0x0f	Wireless alarm ends
0x10	Human body sensing alarm starts
0x11	Human body sensing alarm ends
0x12	Emergency alarm starts
0x13	Emergency alarm ends
0x16	Face detection alarm starts
0x17	Face detection alarm ends
0x1a	Scene change detection alarm
0x1b	Leaving area detection starts
0x1c	Leaving area detection ends
0x1d	Wandering detection starts
0x1e	Wandering detection ends

ISUPSDK (General) Development Guide

value	describe
0x20	Cross-border detection starts
0x21	Cross-border detection ends
0x22	Area intrusion detection starts
0x23	Area intrusion detection alarm ends
0x24	Audio Loss Detection
0x25	Audio anomaly detection
0x26	Defocus detection starts
0x27	Defocus detection ends
0x29	Face detection starts
0x2a	Entering area detection starts
0x2b	Entering area detection ends
0x2c	Crowd detection begins
0x2d	Crowd detection ended
0x2e	Fast motion detection starts
0x2f	Fast motion detection ends
0x30	Face detection completed
0x31	Scene change detection starts
0x32	Scene change detection ends
0x33	Audio loss detection starts
0x34	Audio loss detection ends
0x35	Sound intensity mutation detection begins
0x36	Sound intensity mutation detection ends
0x39	Sound intensity drop detection
0x3c	Parking detection starts
0x3d	Parking detection ends
0x3e	Detection of items left behind begins
0x3f	Detection of items left behind ends
0x40	Item removal detection starts

ISUPSDK (General) Development Guide

value	describe
0x41	Item removal detection ends
0x46	Automatic license plate recognition starts
0x47	Automatic license plate recognition ends
0x48	Fire point detection starts
0x49	Fire point detection completed
0x50	Tamper detection starts
0x51	Anti-tamper detection ends
0x5a	Vessel Detection
0x5b	Temperature warning starts
0x5c	Temperature warning ends
0x5d	Temperature alarm starts
0x5e	Temperature alarm ends
0x5f	Temperature difference alarm starts
0x60	Temperature difference alarm ends
0x400	Zone short circuit alarm
0x401	Zone break alarm
0x402	Abnormal alarm in defense zone
0x403	Zone alarm recovery
0x404	Equipment tampering alarm
0x405	Equipment anti-tampering alarm recovery
0x406	Card reader tamper alarm
0x407	Card reader tamper alarm recovery
0x408	Alarm Input Alarm
0x409	Alarm input Alarm recovery
0x40a	Hostage alarm
0x40b	Memory full alarm
0x40c	Card reading failure alarm

ISUPSDK (General) Development Guide

value	describe
0x411	POS enabled
0x412	POS disabled

Exception log

value	describe
0x20	Disk array abnormality
0x21	Video Loss
0x22	Illegal login
0x23	Hard disk full
0x24	Hard Drive Error
0x25	Modem dropped (hold)
0x26	IP address conflicts
0x27	Network connection lost
0x28	Abnormal video recording
0x29	Network camera connection abnormality
0x2a	Video input abnormality (only applicable to analog channels)
0x2b	IP address conflict of network camera
0x2c	Abnormal scene
0x2d	Capture error Failed to obtain the image
0x2e	Video format does not match
0x2f	The decoding resolution does not match the camera resolution
0x3a	RS-485 serial port connection abnormality
0x3b	RS-485 serial port connection restored
0x3c	Abnormal startup of the daughter board
0x3d	Daughterboard inserted
0x3e	The daughterboard has been removed
0x3f	Abnormal temperature

ISUPSDK (General) Development Guide

value	describe
0x41	Buffer overflow
0x42	Digital signal processing anomaly
0x43	Failed to resume video upload after network disconnection
0x44	Hot standby operation abnormality
0x45	Failed to start the network camera MAS
0x46	IPCM restart abnormality
0x47	PoE power supply abnormality
0x48	Cloud storage data upload failed
0x49	Dial abnormality
0x50	Device offline
0x51	Remote upgrade failed
0x52	Audio loss
0x53	Network camera password synchronization abnormality
0x54	EZVIZ device offline
0x57	Accessory board abnormality
0x400	Device startup
0x401	Device shutdown
0x402	Watchdog recovery
0x403	Low battery
0x404	Battery voltage recovery
0x405	AC disconnect
0x406	AC power restored
0x407	Network recovery connection
0x408	Flash memory read and write abnormality
0x409	Card reader offline
0x40a	Card reader offline recovery
0x4000	Daughterboard IP address conflict
0x4001	Daughterboard offline

value	describe
0x4002	Fan abnormality
0x4003	Abnormal back panel temperature
0x4004	SD card is not healthy
0x4005	SD card is damaged

Operation log

value	describe
0x41	Power on
0x42	Shutdown
0x43	Abnormal shutdown
0x44	Restart the device (locally)
0x50	Login (Local)
0x51	Logout (local)
0x52	Configuration (local)
0x53	Playback or download by file (local)
0x54	Playback or download by time (local)
0x55	Start recording (local)
0x56	Stop recording (local)
0x57	PTZ control (local)
0x58	Preview (local, keep)
0x59	Edit time (local, preserved)
0x5a	Upgrade (local)
0x5b	Video file backup (local)
0x5c	Hard disk initialization (local)
0x5d	Exporting a local configuration file
0x5e	Importing a local configuration file

ISUPSDK (General) Development Guide

value	describe
0x5f	File backup (local)
0x60	Lock video file (local)
0x61	Unlock video files (local)
0x62	Manual clear and trigger alarms (local)
0x63	Adding a network camera (local)
0x64	Delete Network Camera (Local)
0x65	Setting up the network camera (local)
0x66	Start backup (local)
0x67	Stop backup (local)
0x68	Local backup start time
0x69	Local backup end time
0x6a	Add a network hard disk (local)
0x6b	Delete NAS (Local)
0x6c	Setting up NAS (Local)
0x70	Login (remote)
0x71	Logout (remote)
0x72	Start recording (remotely)
0x73	Stop recording (remotely)
0x74	Start Transfer
0x75	Stop Transmission
0x76	Get parameters (remote)
0x77	Remote Configuration
0x78	Get Status (Remote)
0x79	Arming (remote)
0x7a	Disarm (remote)
0x7b	Restart (remote)

ISUPSDK (General) Development Guide

value	describe
0x7c	Start voice intercom
0x7d	Stop voice intercom
0x7e	Remote upgrade
0x7f	Playback by file (remote)
0x80	Playback by time (remote)
0x81	PTZ control (remote)
0x82	Formatting a hard disk (remotely)
0x83	Shutdown (remote)
0x84	Lock File (Remote)
0x85	Unlock file (remotely)
0x86	Export Configuration File (Remote)
0x87	Importing a configuration file (remote)
0x88	Export video file (remote)
0x89	Manual clear and trigger alarm (remote)
0x8a	Adding a network camera (remote)
0x8b	Deleting a network camera (remotely)
0x8c	Setting up the IP camera (remote)
0x8d	Restart the smart library
0x8e	Add NAS (remote)
0x8f	Deleting NAS (remotely)
0x90	Setting up NAS (remotely)
0x91	Start burning (local)
0x92	Stop burning (local)
0x93	Start burning (remote)
0x94	Start burning (remote)
0x95	Image backup (local)

ISUPSDK (General) Development Guide

value	describe
0x96	Image backup (remote)
0x97	Court event resumption (local)
0x98	Court event resumption (remote)
0x99	Import file (local)
0x9a	Delete abnormal or non-existent hard disk
0x9b	Mount hard disk (remote)
0x9c	Unmount the hard disk (remotely)
0x9d	Lock (Local)
0x9e	Unlock (Local)
0x9f	Delete court files (local)
0xd0	Bypass (remote)
0xd1	Bypass recovery (remote)
0xd2	Setting alarm input parameters (remote)
0xd3	Get alarm input parameters (remote)
0xd4	Setting alarm output parameters (remote)
0xd5	Get alarm output parameters (remote)
0xd6	Manually open the alarm output (remote)
0xd7	Manually close the alarm output (remote)
0xd8	Enable or disable the RS-485 serial port of the alarm panel (remote)
0xd9	Export database records
at 0x	Importing Database Records
0xdb	Cascade switching
0xdc	Cascade PTZ control
0x101	Configuring Automatic Rebuild (Local)
0x102	Configuring Hot Standby (Local)
0x103	Create an array (local)

ISUPSDK (General) Development Guide

value	describe
0x104	Delete Array (Local)
0x105	Migrate Array (Local)
0x106	Manually rebuild the array (local)
0x107	One-click configuration (local)
0x108	Create a virtual disk (local)
0x109	Deleting a virtual disk (local)
0x10a	Repair Virtual Disk (Local)
0x10b	Expand virtual disk (local)
0x10c	Upgrading a disk array (local)
0x10d	Safely remove the disk array (local)
0x111	Configuring automatic rebuild (remote)
0x112	Configuring hot standby (remote)
0x113	Create Array (Remote)
0x114	Deleting an Array (Remotely)
0x115	Migrate Array (Remote)
0x116	Manually rebuilding the array (remotely)
0x117	One-click configuration (remote)
0x118	Create a virtual disk (remote)
0x119	Deleting a virtual disk (remotely)
0x11a	Repair Virtual Disk (Remote)
0x11b	Expand Virtual Disk (Remote)
0x11c	Upgrading a disk array (remotely)
0x11d	Safely remove the disk array (remotely)
0x121	Start capturing (local)
0x122	Stop capturing (local)
0x125	Setting up SNMP (Local)

ISUPSDK (General) Development Guide

value	describe
0x126	Tag Operations (Local)
0x131	Start capturing (remote)
0x132	Stop capturing (remote)
0x135	Setting up SNMP (remote)
0x136	Tag Operation (Remote)
0x140	Switching output port (local)
0x141	Configuring Encoding Performance
0x142	N+1 Hot Standby Operation (Local)
0x143	N+1 hot standby operation (remote)
0x144	Exporting a network camera configuration file (local)
0x145	Importing a network camera configuration file (local)
0x146	Upgrading Network Camera (Local)
0x147	Exporting a network camera configuration file (remote)
0x148	Importing a network camera configuration file (remote)
0x149	Upgrading IP Camera (Remote)
0x201	Set the main screen of the multi-screen controller
0x202	Set up the sub-screen of the multi-screen controller
0x203	Close the main screen of the multi-screen controller
0x204	Close the sub-screen of the multi-screen controller
0x251	Modify input source
0x252	Modify output channel
0x253	Modify the virtual LED
0x254	Modify the logo
0x255	Setting the scene
0x256	Display Operation
0x257	Get Virtual LED
0x258	Get the scene

ISUPSDK (General) Development Guide

value	describe
0x259	Scene Control
0x260	Get all available window information
0x261	Get single window information
0x262	Window Control
0x263	Get scene list
0x264	Scene Control
0x265	Setting up a single scene
0x266	Get the input source list
0x267	Get the plan list
0x268	Edit plan
0x269	Control Plan
0x270	Screen Controls
0x271	Add a signal source
0x272	Edit Source
0x273	Set the decoder board parameters
0x274	Get the decoder board parameters
0x275	Get device information
0x276	Upload background image
0x277	set password
0x278	Add a scene
0x279	Deleting a scene
0x280	Delete a source
0x281	Add a plan
0x282	Delete plan
0x283	Get external matrix parameters
0x284	Setting external matrix parameters
0x285	Get user parameters
0x286	Setting User Parameters

ISUPSDK (General) Development Guide

value	describe
0x287	Get TV wall connection parameters
0x288	Set the TV wall connection parameters
0x289	Get TV wall scene information
0x28a	Set the TV wall scene information
0x28b	Get current scene information
0x28c	Scene Switching
0x300	Mount hard disk (local)
0x301	Deletion abnormality or non-existent hard disk (local)
0x400	Open door (remote)
0x401	Close door (remote)
0x402	Door normally open (remote)
0x403	Door normally closed (remote)
0x404	Manual time adjustment (remote)
0x405	Automatic NTP time synchronization
0x406	Clear card number (remote)
0x407	Restore factory settings (remotely)
0x408	Zone Arming
0x409	Zone disarming
0x40a	Factory Reset (Local)
0x1001	Set the trigger mode
0x1002	Get trigger mode
0x1003	Set alarm output parameters
0x1004	Get alarm output parameters
0x1005	Get recommended trigger mode
0x1006	Obtaining Condition Monitoring Parameters
0x1007	Setting Condition Monitoring Parameters
0x1008	Obtaining Condition Monitoring Parameters

ISUPSDK (General) Development Guide

value	describe
0x1009	Get video alarm mode
0x100a	Set the video alarm mode
0x2001	Add vehicle information (local)
0x2002	Edit vehicle information (local)
0x2003	Delete vehicle information (local)
0x2004	Search vehicle information (local)
0x2005	Adding arming information (local)
0x2006	Editing Arming Information (Local)
0x2007	Deleting the arming information (local)
0x2008	Search for arming information (local)
0x2009	Search for normal traffic information (local)
0x200a	Search for unusual traffic information (local)
0x200b	Search for normal traffic information (local)
0x200c	Preview image (local)
0x200d	Setting entry and exit parameters (local)
0x200e	Get entry and exit parameters (local)
0x200f	Set data upload parameters (local)
0x2010	Get data upload parameters (local)
0x2011	Open or close the gate
0x2012	Add peripheral information (local)
0x2013	Edit peripheral information (local)
0x2014	Delete peripheral information (local)
0x2015	Search for peripheral information (local)
0x2016	Add parking rules (local)
0x2017	Edit parking rules (local)
0x2018	Delete parking rules (local)

ISUPSDK (General) Development Guide

value	describe
0x2019	Search parking rules (local)
0x2020	Normal passing data statistics (local)
0x2021	Export normal traffic statistics report (local)
0x2022	Abnormal traffic statistics (local)
0x2023	Export abnormal traffic statistics report (local)
0x2024	Pedestrian traffic statistics
0x2025	Export pedestrian traffic statistics report (local)
0x2026	Search vehicle toll information (local)
0x2027	Vehicle toll statistics (local)
0x2028	Export vehicle toll statistics report (local)
0x2029	Search shift information (local)
0x2030	Search card information (local)
0x2031	Add discount rules (local)
0x2032	Edit discount rules (local)
0x2033	Delete discount rule (local)
0x2034	Search discount rules (local)
0x2035	Get the offline detection parameters of the access control machine (local)
0x2036	Set the access control machine offline detection parameters (local)
0x2037	Set the card information issued by the access control machine (local)
0x2038	Clear the card information issued by the access control machine (local)
0x2101	Open or close the gate (remotely)
0x2102	Set entry and exit parameters (remote)
0x2103	Get entry and exit parameters (remote)
0x2104	Set data upload parameters (remote)
0x2105	Get data upload parameters (remote)
0x2106	Get basic terminal information (remote)

ISUPSDK (General) Development Guide

value	describe
0x2107	Get character overlay parameters (remote)
0x2108	Set character overlay parameters (remote)
0x2109	Get intersection information (remote)
0x210a	Establishing a data synchronization server (remote)
0x210b	Get the working status of the entry and exit terminals (remote)
0x210c	Get the status of the entry and exit terminal channel (remote)
0x210d	Add peripheral information (remote)
0x210e	Edit peripheral information (remote)
0x210f	Get the offline detection parameters of the access control machine (remote)
0x2110	Set the access control machine offline detection parameters (remote)
0x2111	Set the card information issued by the access control machine (remote)
0x2112	Clear the card information issued by the access control machine (remote)
0x2115	Turn on parking indicator light control (remote)
0x2116	Disable parking indicator control (remote)
0x2117	Setting the volume
0x2118	Set the recording volume
0x2119	Intelligent configuration
0x211a	Set disk array speed (local)
0x211b	Set disk array speed (remote)
0x211c	Adding a storage pool (remote)
0x211d	Deleting a Storage Pool (Remote)
0x2120	Delete image (remotely)
0x2121	Delete recording (remotely)
0x2123	Enable cloud storage (remote)
0x2124	Disable cloud storage (remote)
0x2125	Edit cloud storage pool parameters (remote)

value	describe
0x2126	Edit cloud storage pool capacity (remotely)
0x2200	Create or edit a view gallery space (remotely)
0x2201	Delete view library files (remote)
0x2202	Download view library files (remote)
0x2203	Upload view library files (remote)
0x2204	Create or edit a view library space (local)
0x2501	MCU Conference Configuration
0x2502	MCU terminal configuration
0x2503	MCU Group Configuration
0x2504	MCU Conference Control
0x2505	MCU terminal control

B.5.1 Alarm host log type

Alarm log

Log Type	describe
0x01	Short circuit alarm
0x02	Short circuit alarm
0x03	alarm reset
0x04	Alarm recovery
0x05	The password is incorrect for three consecutive times
0x06	Invalid card number
0x07	Keyboard tamper alarm
0x08	Keyboard anti-tamper recovery
0x09	Equipment tampering alarm

ISUPSDK (General) Development Guide

Log Type	describe
0x0a	Equipment anti-tampering recovery
0x0b	The analog value is lower than the alarm limit 1
0x0c	The analog value is lower than the alarm limit 2
0x0d	The analog value is lower than the alarm limit 3
0x0e	The analog value is lower than the alarm limit 4
0x0f	The analog value is higher than the alarm limit 1
0x10	The analog value is higher than the alarm limit 2
0x11	The analog value is higher than the alarm limit 3
0x12	The analog value is higher than the alarm limit 4
0x13	Emergency button pressed
0x14	Emergency button reset
0x15	Soft defense zone police
0x16	Soft zone fire alarm
0x17	Soft zone emergency alarm
0x18	Motion detection alarm starts
0x19	Motion detection alarm ends
0x1a	Device blocking
0x1b	Device recovery
0x1c	UPS Alarm
0x1d	Electricity meter alarm
0x1e	Power switch alarm
0x1f	Gas detection alarm
0x20	Transformer temperature alarm

ISUPSDK (General) Development Guide

Log Type	describe
0x21	Temperature and humidity sensor alarm
0x22	UPS Recovery
0x23	Power meter recovery
0x24	Power switch restored
0x25	Gas detection recovery
0x26	Transformer temperature recovery
0x27	Temperature and humidity sensor recovery
0x28	Temperature and humidity sensor recovery
0x29	Water level sensor recovery
0x2a	Dust and noise sensor alarm
0x2b	Dust and noise sensor recovery
0x2c	Environmental collector alarm
0x2d	Environmental collector recovery
0x2e	Detector tamper alarm
0x2f	Detector recovery
0x30	Emergency Alarm
0x31	Emergency alarm reset

Exception log

Log Type	describe
0x01	Power on
0x02	Shutdown
0x03	WDT Reset
0x04	Low battery voltage

ISUPSDK (General) Development Guide

Log Type	describe
0x05	AC power failure
0x06	AC power restored
0x07	RTC real-time clock abnormality
0x08	Network connection lost
0x09	Network connection restored
0x0a	The phone line is disconnected
0x0b	Phone line link restored
0x0c	Expansion bus disconnected
0x0d	Expansion bus connection restored
0x0e	Keyboard bus connection lost
0x0f	Keyboard bus connection recovery
0x10	Simulating sensor failure
0x11	Simulate sensor recovery
0x12	RS-485 serial port connection disconnected
0x13	RS-485 serial port connection restored
0x14	Battery voltage recovery
0x15	Wired network abnormality
0x16	Wired network recovery
0x17	GPRS abnormality
0x18	GPRS Recovery
0x19	3G network abnormality
0x1a	3G network recovery
0x1b	SIM card abnormality

ISUPSDK (General) Development Guide

Log Type	describe
0x1c	SIM Card Recovery
0x1d	Video Loss
0x1e	Illegal login
0x1f	Hard disk full
0x20	Hard Drive Error
0x21	Modem disconnection
0x22	IP address conflicts
0x23	Network connection lost
0x24	Abnormal video recording
0x25	Video input abnormality (only applicable to analog channels)
0x26	Remote hard disk formatting failed
0x27	USB communication failure
0x28	USB communication recovery
0x29	Printer failure
0x2a	Printer Recovery
0x2b	Daughter board communication error
0x2c	IP camera offline
0x2d	IP address conflict of network camera
0x2e	Video format does not match

Operation log

Log Type	describe
0x01	Normal Arming
0x02	Normal disarm
0x03	bypass
0x04	Hostage
0x05	Local Restart
0x06	Remote Reboot
0x07	Local upgrade
0x08	Remote upgrade
0x09	reset
0x0a	Remote alarm output control
0x0b	Access control: open
0x0c	Door closed
0x0d	Siren on
0x0e	Siren off
0x0f	Defense zone settings
0x10	Alarm output settings
0x11	Sensor Settings
0x12	RS-485 serial port settings
0x13	Dial Settings
0x14	Adding an Admin User
0x15	Editing an Admin User
0x16	Deleting an Admin User

ISUPSDK (General) Development Guide

Log Type	describe
0x17	Add DVR/NVR operation user
0x18	Modify DVR/NVR operation user
0x19	Delete DVR/NVR operation user
0x1a	Add camera operation user
0x1b	Change the password of the camera operation user
0x1c	Deleting the password of the camera operation user
0x1d	Adding a Keyboard/Card Reader User
0x1e	Deleting a Keyboard/Card Reader User
0x1f	Remote login
0x20	Remote Logout
0x21	Remote Arming
0x22	Remote disarming
0x23	Edit alarm panel parameters
0x24	Bypass recovery
0x25	Enable Output
0x26	Disable Output
0x27	Editing Subsystem Parameters
0x28	Group Bypass
0x29	Group bypass recovery
0x2a	Edit GPRS parameters
0x2b	Editing Network Upload Parameters
0x2c	Edit Upload Mode
0x2d	Edit access control parameters

ISUPSDK (General) Development Guide

Log Type	describe
0x2e	Remote: Start recording
0x2f	Remote: Stop recording
0x30	Transmission starts
0x31	End of transmission
0x32	Voice intercom starts
0x33	End of voice talk
0x34	Remote: Playback or download by file
0x35	Remote: Playback by time
0x36	Remote: PTZ control
0x37	Remote: Lock File
0x38	Remote: Unlock file
0x39	Remote: Format hard disk
0x3a	Remote: Export configuration file
0x3b	Remote: Import configuration file
0x3c	Remote: Export File
0x3d	Arm at home
0x3e	Instant Arming
0x3f	Automatic arming
0x40	Automatic disarming
0x41	Key zone arming
0x42	Key zone disarm
0x43	Clear alarm
0x44	Editing system fault parameters

ISUPSDK (General) Development Guide

Log Type	describe
0x45	Edit alarm output parameters
0x46	Search peripheral modules
0x47	Re-register peripheral modules
0x48	Turn off the keyboard alarm tone
0x49	Editing Mobile Network Parameters
0x4a	Edit Printer Parameters
0x4b	Format SD card
0x4c	Upgrade daughter board
0x4d	Arm or disarm schedule configuration
0x4e	SMS Arming
0x4f	Arm your phone at home
0x50	Instant mobile phone deployment
0x51	Mobile phone disarm
0x52	Clear phone alarm
0x53	Authorization list settings
0x54	Enable or disable trigger configuration on a schedule
0x55	Snapshot Settings
0x56	Zone tamper settings
0x57	Remote: Unlock
0x58	Remote: Lock
0x59	Local: Unlock
0x5a	Local: Locked
0x5b	Remote: Turn on the warning lights

Log Type	describe
0x5c	Remote: Turn off the warning light
0xf0	Local: Activate device
0xf1	Remote: Activate device
0xf2	Local: Restore factory settings
0xf3	Remote: Restore factory settings

Event Log

Log Type	describe
0x01	B code synchronization
0x02	SDK Sync
0x03	Adjust time according to plan
0x04	Insert the daughter board
0x05	Remove the daughterboard
0x06	Automatic arming
0x07	Automatic disarming
0x08	Activate triggers on a schedule
0x09	Deactivate triggers by schedule
0x0a	Auto arming failed
0x0b	Automatic disarming failed
0x0c	Failed to activate trigger
0x0d	Deactivation trigger failed
0x0e	Forced Arming

B.5.2 Log parameter configuration

Allows HCISUPSDK to configure log parameters during runtime.

The specific content of the HCISUPSDK_Log_Switch.xml file is as follows:

```
<?xml version="1.0" ncnf8 <SdkLocal><!--
Required, SDK local configuration parameters, outermost node-->
<CMSLog><!--Optional, log configuration parameters of HCISUPCMS
  module--> <logLevel>3</logLevel><!--Required, xs:integer, log level, 1-ERROR, 2-DEBUG, 3-INFO--> <logDirectory>./
  CMSLog/</logDirectory><!--Required, xs:string, log file output directory, which can be a relative path or a
Absolute path, must end with "/" -->
  <autoDelete>true</autoDelete><!--Required, xs:boolean, whether the log file is automatically deleted. When set to true, the module will generate no
more than 10 log files, and the logs will be cyclically overwritten (the first log file does not participate in the cyclic overwriting). When set to false, there is
no upper limit on the
  number of
  logs--> </CMSLog> <StreamLog><!--Optional, log configuration parameters of the
  HCISUPStream module--> <logLevel>3</logLevel><!--Required, xs:integer, log level, 1-ERROR, 2-DEBUG, 3-INFO-->
  > <logDirectory>./StreamLog/</logDirectory><!--Required, xs:string, log file output directory, which can be a relative path or
It is an absolute path and must end with "/" -->
  <autoDelete>true</autoDelete><!--Required, xs:boolean, whether the log file is automatically deleted. When set to true, the module will generate no
more than 10 log files, and the logs will be cyclically overwritten (the first log file does not participate in the cyclic overwriting). When set to false, there is
no upper limit on the
  number of logs-->
  </StreamLog> <AlarmLog><!--Optional, log configuration parameters of the
  HCISUPAlarm module--> <logLevel>3</logLevel><!--Required, xs:integer, log level, 1-ERROR, 2-DEBUG, 3-INFO-->
  <logDirectory>./AlarmLog/</logDirectory><!--Required, xs:string, log file output directory, which can be a relative path or
It is an absolute path and must end with "/" -->
  <autoDelete>true</autoDelete><!--Required, xs:boolean, whether the log file is automatically deleted. When set to true, the module will generate no
more than 10 log files, and the logs will be cyclically overwritten (the first log file does not participate in the cyclic overwriting). When set to false, there is
no upper limit on the
  number of logs-->
  > </AlarmLog> <SSLog><!--Optional, log configuration parameters of the
  HCISUPSS module--> <logLevel>3</logLevel><!--Required, xs:integer, log level, 1-ERROR, 2-DEBUG, 3-INFO-->
  <logDirectory>./SSLog/</logDirectory><!--Required, xs:string, log file output directory, which can be a relative path or an absolute path and must end
with "/"--> <autoDelete>true<!--
  autoDelete><!--Required, xs:boolean, whether the log file is automatically deleted. When set to true, the module will generate no more than 10 log
files. , and perform log cycle coverage (the first log file does not participate in the cycle coverage). When set to false, there is no upper limit on the
  number of logs. --> /
  SSLog> </
SdkLocal>
```

Notes •

The default HCISUPSDK product does not contain HCISUPSDK_Log_Switch.xml. • When the running directory does not contain the HCISUPSDK_Log_Switch.xml configuration file, the module's log behavior is based on the SetLogToFile interface ([NET_ECMS_SetLogToFile](#) , [NET_ESTREAM_SetLogToFile](#) , [NET_EALARMS_SetLogToFile](#) and [NET_ESS_SetLogToFile](#) shall prevail.

- When the running directory contains the HCISUPSDK_Log_Switch.xml configuration file, but the HCISUPSDK_Log_Switch.xml

When the XML format of the configuration file is incorrect and HCISUPSDK cannot parse it properly, the module's logging behavior is to call `SetLogToFile` file (`NET_ECMS_SetLogToFile` & `NET_ESTREAM_SetLogToFile`).

The log parameters set by `NET_EALARM_SetLogToFile` and `NET_ESS_SetLogToFile` shall prevail.
 - When the running directory contains the HCISUPSDK_Log_Switch.xml configuration file, but the HCISUPSDK_Log_Switch.xml

The configuration file is correct, and the configuration in the configuration file shall prevail. If the configuration file does not contain the configuration of a certain module (for example, `<SSLLog>`), the logging behavior of the module (such as HCISUPSS module) is set by calling the `SetLogToFile` interface.

The log parameters shall prevail.
 - When the running directory contains the HCISUPSDK_Log_Switch.xml configuration file, but HCISUPSDK_Log_Switch.xml

If the format is correct but the parameters of a module are incorrect, the logging behavior of the module is based on the interface, and other correct modules are based on the configuration.

The file configuration shall prevail.
 - When the running directory contains the HCISUPSDK_Log_Switch.xml configuration file, but HCISUPSDK_Log_Switch.xml

The format and parameters are correct. But then the configuration file is modified and becomes corrupted. Then the module log line

For the default, the last correct configuration (configuration file or interface, but configuration file takes precedence) is used.
 - When the log directory of the cause is modified through the log configuration file, the original log directory and its log files will not be changed.
- New logs are printed to a new directory, which is equivalent to calling the log setting interface to adjust the log directory.

B.6 Supported IP Cameras

Product Type and Series	Product number	Encrypted Index RS-485	
50 Series Bolt Action	DS-2CD5046FWD(-ŷ-AP)	DS-2CD5046FWD Support	
	DS-2CD5065FWD(-ŷ-AP)	DS-2CD5065FWD Support	
	DS-2CD5085FWD	DS-2CD5085FWD Support	
	DS-2CD5027FWD-A(B)	DS-2CD5027FWD(B) supports	
	DS-2CD5046F-A(B)	DS-2CD5046F(B) supports	
	DS-2CD5065F(B)	DS-2CD5065F(B) supports	
	DS-2CD5086F(B)	DS-2CD5086F(B) supports	
	DS-2CD5046F-A(C)	DS-2CD5046F(C) supports	
	DS-2CD5065F-A(C)	DS-2CD5065F(C) Support	
	DS-2CD5085F(C)	DS-2CD5085F(C) Support	

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
51 Series Indoor Dome DS-2CD5166FWD-IZ (2.8~12mm)	DS-2CD5166FWD-IZ Support	
	DS-2CD5186FWD-IZ(2.8~12mm)	DS-2CD5186FWD-IZ Support
5A Series Roller Machine	DS-2CD5A27FWD-IZHS(2.8-12mm)(B)	DS-2CD5A27FWD(B) supports
	DS-2CD5A27FWD-IZHS(8-32mm)(B)	DS-2CD5A27FWD(B) supports
	DS-2CD5A46F-IZHS(2.8-12mm)(B)	DS-2CD5A46F(B) supports
	DS-2CD5A65F-IZHS(2.8-12mm)(B)	DS-2CD5A65F(B) supports
	DS-2CD5A86F-IZHS(2.8-16mm)(B)	DS-2CD5A86F(B) supports
	DS-2CD5A46FWD(-IZHSÿ-IZSÿ-IZ)	DS-2CD5A46FWD Support
	DS-2CD5A52FWD(-IZHSÿ-IZSÿ-IZ)	DS-2CD5A52FWD Support
	DS-2CD5A65FWD(-IZHSÿ-IZSÿ-IZ)	DS-2CD5A65FWD Support
	DS-2CD5A85FWD(-IZHSÿ-IZSÿ-IZ)	DS-2CD5A85FWD Support
	DS-2CD5A46F-IZHS(C)	DS-2CD5A46F(C) Support
	DS-2CD5A52F-IZHS(C)	DS-2CD5A52F(C) Support
	DS-2CD5A65F-IZHS(C)	DS-2CD5A65F(C) Support
	DS-2CD5A85F-IZHS(C)	DS-2CD5A85F(C) Support
70 Series Bolt Action	DS-2CD7027EWD	DS-2CD7027EWD Support
	DS-2CD7047EWD	DS-2CD7047EWD Support
	DS-2CD7067EWD	DS-2CD7067EWD Support
	DS-2CD70C7EWD-A (domestic standard) (white)	DS-2CD70C7EWD Support
	DS-2CD70C7EWD (domestic standard) (white)	DS-2CD70C7EWD Support
	DS-2CD7087EWD	DS-2CD7087EWD Support
71 Series Hemisphere	DS-2CD7127EWD-IZ(2.8-12mm)(8-32mm)	DS-2CD7127EWD Support

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
	DS-2CD7147EWD-IZ(2.8-12mm)	DS-2CD7147EWD Support
	DS-2CD7167EWD-IZ(2.8-12mm)	DS-2CD7167EWD Support
	DS-2CD7187EWD-IZ(2.8-12mm)	DS-2CD7187EWD Support
	DS-2CD7147EWD-IZ(8-32mm)	DS-2CD7147EWD Support
	DS-2CD7167EWD-IZ(8-32mm)	DS-2CD7167EWD Support
	DS-2CD71C7EWD-IZ(2.8-12mm)(domestic standard) (white)	DS-2CD71C7EWD Support
	DS-2CD7187EWD-IZ(8-32mm)	DS-2CD7187EWD Support
7A Series Roller Machine	DS-2CD7A27EWD/F-(I)(L)Z(S)(2.8-12mm) (8-32mm)	DS-2CD7A27EWD Support
	DS-2CD7A47EWD-(I)(L)Z(S)(2.8-12mm)	DS-2CD7A47EWD jy
	DS-2CD7A67EWD-(I)(L)Z(S)(2.8-12mm)	DS-2CD7A67EWD jy
	DS-2CD7A87EWD-(I)(L)Z(S)(2.8-12mm)	DS-2CD7A87EWD jy
	DS-2CD7AC7EWD-XZ(S)(2.8-12mm)(Domestic standard) (white)	DS-2CD7AC7EWD Support
	DS-2CD7AC7EWD-XZ(S)/ZJ(2.8-12mm)(domestic standard) (white)	DS-2CD7AC7EWD Support
	DS-2CD7A27EWD-XZ(S)(2.8-12mm) (8-32mm)	DS-2CD7A27EWD-X Support
	DS-2CD7A47EWD-XZ(S)(2.8-12mm) (8-32mm)	DS-2CD7A47EWD-X Support
	DS-2CD7A67EWD-XZ(S)(2.8-12mm) (8-32mm)	DS-2CD7A67EWD-X Support
	DS-2CD7A87EWD-XZ(S)(2.8-12mm) (8-32mm)	DS-2CD7A87EWD-X Support

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
7D Series Dome	DS-2CD7D27DWD-IZS(W)(4mm-6mm) (6mm-9mm)	DS-2CD7D27DWD not supported
	DS-2CD7D27DWD-IZS(4-6mm)	DS-2CD7D27DWD not supported
	DS-2CD7D47DWD-IZS(6-9mm)	DS-2CD7D47DWD not supported
	DS-2CD7D47DWD-IZS(W)(4mm-6mm) (6mm-9mm)	DS-2CD7D47DWD not supported
7T Series Tube Machine	DS-2CD7T27DWD-IZS(GL)(W)(4mm-6mm)(6mm-9mm)	DS-2CD7T27DWD not supported
	DS-2CD7T27DWD-IZSW(6-9mm)(B)	DS-2CD7T27DWD-B not supported
	DS-2CD7T27DWD-IZSW(4-6mm)(B)	DS-2CD7T27DWD-B not supported
	DS-2CD7T47DWD-IZSW(6-9mm)(B)	DS-2CD7T47DWD-B not supported
	DS-2CD7T47DWD-IZSW(4-6mm)(B)	DS-2CD7T47DWD-B not supported
	DS-2CD7T27DWD-IZS(6-9mm)(B)	DS-2CD7T27DWD-B not supported
	DS-2CD7T27DWD-IZS(4-6mm)(B)	DS-2CD7T27DWD-B not supported
	DS-2CD7T47DWD-IZS(6-9mm)(B)	DS-2CD7T47DWD-B not supported
	DS-2CD7T47DWD-IZS (4-6mm)(B)	DS-2CD7T47DWD-B not supported
	DS-2CD7T27DWD-IZS/POE (6-9mm)(B) DS-2CD7T27DWD-B yyÿ	
	DS-2CD7T27DWD-IZS/POE (4-6mm)(B) DS-2CD7T27DWD-B Not supported	
	DS-2CD7T47DWD-IZS/POE (6-9mm)(B) DS-2CD7T47DWD-B Not supported	
	DS-2CD7T47DWD-IZS/POE (4-6mm)(B) DS-2CD7T47DWD-B Not supported	
	DS-2CD7T47DWD-IZS(GL)(W)(4mm-6mm)(6mm-9mm)	DS-2CD7T47DWD not supported
70 Series Bolt Camera (H3+MA)	DS-2CD7027FWD/VA(/NMFC)(/DN)	DS-2CD7027FWD/V Support

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
	DS-2CD7047F/V-A(/NMFC)(/DN)	DS-2CD7047F/V Support
	DS-2CD7067F/V-A(/NMFC)(/DN)	DS-2CD7067F/V Support
	DS-2CD7087F/V(/NMFC)(/DN)	DS-2CD7087F/V Support
	DS-2CD7026FWD-A(/NMFC)(/DN)	DS-2CD7026FWD Support
	DS-2CD7046F-A(/NMFC)(/DN)	DS-2CD7046F Support
	DS-2CD7066F-A(/NMFC)(/DN)	DS-2CD7066F Support
	DS-2CD7086F(/NMFC)(/DN)	DS-2CD7086F Support
	DS-2CD7046FWD(-, -A)(/NMFC)(/DN)	DS-2CD7046FWD supports
	DS-2CD7066FWD(-, -A)(/NMFC)(/DN)	DS-2CD7066FWD supports
	DS-2CD7086FWD(/NMFC)(/DN)	DS-2CD7086FWD Support
	DS-2CD7047FWD/V(/NMFC)(/DN)	DS-2CD7047FWD/V Support
	DS-2CD7067FWD/V(/NMFC)(/DN)	DS-2CD7067FWD/V Support
	DS-2CD7087FWD/V(/NMFC)(/DN)	DS-2CD7087FWD/V Support
	DS-2CD7027FWD/F-A(/NMFC)(/DN)	DS-2CD7027FWD/F Support
	DS-2CD7027FWD-A(/NMFC)(/DN)	DS-2CD7027FWD Support
	DS-2CD7047F/F-A(/NMFC)(/DN)	DS-2CD7047F/F Support
	DS-2CD7047FWD-A(/NMFC)(/DN)	DS-2CD7047FWD Support
	DS-2CD7067F/F-A(/NMFC)(/DN)	DS-2CD7067F/F Support
	DS-2CD7067FWD-A(/NMFC)(/DN)	DS-2CD7067FWD Support
	DS-2CD7087F/F-A(/NMFC)(/DN)	DS-2CD7087F/F Support
	DS-2CD7087FWD-A(/NMFC)(/DN)	DS-2CD7087FWD Support
	DS-2CD7126FWD-IZ(2.8-12mm)(8-32mm)	DS-2CD7126FWD supports

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
	DS-2CD7127FWD/F-IZ(2.8-12mm) (8-32mm)	DS-2CD7127FWD Support
	DS-2CD7127FWD/H-IZ(2.8-12mm)	DS-2CD7127FWD/H Support
	DS-2CD7147FWD-IZ(2.8-12mm)	DS-2CD7147FWD Support
	DS-2CD7167FWD-IZ(2.8-12mm)	DS-2CD7167FWD Support
	DS-2CD7187FWD-IZ(2.8-12mm)	DS-2CD7187FWD Support
	DS-2CD7147FWD-IZ(8-32mm)	DS-2CD7147FWD Support
	DS-2CD7167FWD-IZ(8-32mm)	DS-2CD7167FWD Support
	DS-2CD7187FWD-IZ(8-32mm)	DS-2CD7187FWD Support
7A Series Drum Machine (H3-MA)	DS-2CD7A27FWD/V-(I)(L)ZS(2.8-12mm)	DS-2CD7A27FWD/V Support
	DS-2CD7A27FWD/V-(I)(L)ZS(8-32mm)	DS-2CD7A27FWD/V Support
	DS-2CD7A47F/V-(I)(L)ZS(2.8-12mm)	DS-2CD7A47F/V Support
	DS-2CD7A67F/V-(I)(L)ZS(2.8-12mm)	DS-2CD7A67F/V Support
	DS-2CD7A87F/V-(I)(L)ZS(2.8-16mm)	DS-2CD7A87F/V Support
	DS-2CD7A26FWD-(I)(L)ZS(2.8-12mm)	DS-2CD7A26FWD Support
	DS-2CD7A26FWD-(I)(L)ZS(8-32mm)	DS-2CD7A26FWD Support
	DS-2CD7A46F-(I)(L)ZS(2.8-12mm)	DS-2CD7A46F Support
	DS-2CD7A66F-(I)(L)ZS(2.8-12mm)	DS-2CD7A66F Support
	DS-2CD7A86F-(I)(L)ZS(2.8-16mm)	DS-2CD7A86F Support
	DS-2CD7A46FWD(-IZSÿ-IZ)	DS-2CD7A46FWD Support
	DS-2CD7A66FWD(-IZSÿ-IZ)	DS-2CD7A66FWD Support
	DS-2CD7A86FWD(-IZSÿ-IZ)	DS-2CD7A86FWD Support
	DS-2CD7A47FWD/V	DS-2CD7A47FWD/V Support

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
	DS-2CD7A67FWD/V(-IZSÿ-IZ)	DS-2CD7A67FWD/V Support
	DS-2CD7A87FWD/V(-IZSÿ-IZ)	DS-2CD7A87FWD/V Support
	DS-2CD7A27FWD/F-(L)(I)ZS(2.8-12mm) DS-2CD7A27FWD/F yy	
	DS-2CD7A27FWD/F-(L)(I)ZS(8-32mm)	DS-2CD7A27FWD/F Support
	DS-2CD7A47F/F-IZS(2.8-12mm)	DS-2CD7A47F/F support
	DS-2CD7A47FWD/F-IZS(2.8-12mm)(/NMFC)(/DN)	DS-2CD7A47FWD Support
	DS-2CD7A67F/F-IZS(2.8-12mm)	DS-2CD7A67F/F Support
	DS-2CD7A67FWD-IZS(2.8-12mm)(/NMFC)(/DN)(8-32mm)	DS-2CD7A67FWD Support
	DS-2CD7A87F/F-IZS(2.8-16mm)	DS-2CD7287F/F Support
	DS-2CD7A87FWD-IZS(2.8-16mm)(/NMFC)(/DN)(8-32mm)	DS-2CD7A87FWD Support
	DS-2CD7A27FWD-XZ(S)(2.8-12mm)(8-32mm)	DS-2CD7A27FWD-X Support
	DS-2CD7A47FWD-XZ(S)(2.8-12mm)(8-32mm)	DS-2CD7A47FWD-X Support
	DS-2CD7A67FWD-XZS(2.8-12mm)(8-32mm)	DS-2CD7A67FWD-X Support
	DS-2CD7A87FWD-XZS(2.8-12mm)(8-32mm)	DS-2CD7A87FWD-X Support
72 Series Bullet Camera (H3+MA)	DS-2CD7287F/V (5.7-21mm, 15 - 55mm manual zoom and electric zoom)	DS-2CD7287F/V Support
	DS-2CD72C7FWD/V (25mm, 30mm, 50mm fixed focal length lens)	DS-2CD72C7FWD/V Support

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
	DS-2CD7287F/F	DS-2CD7287F/F Support
	DS-2CD72C7FWD/F	DS-2CD72C7FWD/F Support
78 Series Drum Machine (H3+MA)	DS-2CD7826FWD-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)(8-32mm)	DS-2CD7826FWD Support
	DS-2CD7827FWD/V-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)(8-32mm)	DS-2CD7827FWD/V Support
	DS-2CD7827FWD/F-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)(8-32mm)	DS-2CD7827FWD/F Support
	DS-2CD7846FWD-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)	DS-2CD7846FWD Support
	DS-2CD7847FWD/V-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)	DS-2CD7847FWD/V Support
	DS-2CD7847F/F-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)	DS-2CD7847F/F Support
	DS-2CD7866FWD-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)	DS-2CD7866FWD Support
	DS-2CD7867FWD/V-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)	DS-2CD7867FWD/V Support
	DS-2CD7867F/F-(I)(L)Z(S)(/SWG)(/GLG) (2.8-12mm)	DS-2CD7867F/F Support
	DS-2CD7886F-(I)(L)Z(S)(/SWG)(/GLG) (2.8-16mm)	DS-2CD7886F Support
	DS-2CD7887F/V-(I)(L)Z(S)(/SWG)(/GLG) (2.8-16mm)	DS-2CD7887F/V Support
	DS-2CD7887F/F-(I)(L)Z(S)(/SWG)(/GLG) (2.8-16mm)	DS-2CD7887F/F support

ISUPSDK (General) Development Guide

Product Type and Series	Product number	Encrypted Index RS-485
Protective cover integrated machine (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)	/
Explosion-proof face sensor (H3+MA) DS-2XE8427FWD-LZHS (8-32mm ZF2-D) (2.8-12mm ZF11)	DS-2XE8427FWD Support	
	DS-2XE8447FWD-(L)(I)ZHS(8-32mm ZF15)	DS-2XE8447FWD ſü
	DS-2XE8487FWD-(L)(I)ZHS(8-32mm ZF15)	DS-2XE8487FWD ſü

B.7 Supported Network Video Recorders**I** Series Network Video Recorder

Product number
DS-9664N-I16
DS-9664N-I8
DS-9616N-I8(8*8T AI disk)
DS-9632N-I8
DS-9616N-I8
DS-9608N-I8
DS-9616N-I16
DS-9632N-I16

Product number
DS-FN64DGX(B)
DS-9664N-I16(16*8T AI disk)
DS-FN16DGX(B)
DS-9616N-I16(16*8T AI disk)
DS-FN32DGX(B)
DS-9632N-I16(16*8T AI disk)
DS-9632N-I8(8*8T AI disk)
DS-8632N-I8/24P
DS-8616N-I8
DS-FN32CGX(B)
DS-8608N-I8(8*6T)
DS-8616N-I8(8*6T)
DS-8632N-I8
DS-8664N-I8
DS-FN08CGX(B)
DS-8664N-I16
DS-FN08MG
DS-8616N-I16
DS-8632N-I8(8*6T)
DS-8608N-I8
DS-8664N-I8/MB-84
DS-FN16CGX(B)

ISUPSDK (General) Development Guide

Product number
DS-8632N-I16(16*6T)
DS-FN16DG(C)
DS-FN08PG(B)
DS-8632N-I16
DS-FN64CG(C)
DS-FN16MG
DS-FN16PG(B)
DS-8632N-I8/MB-84
DS-FN32DG(C)
DS-8616N-I8/MB-84
DS-8664N-I8(8*6T)
DS-GJZ9N08-F86(B)
DS-FN32MG
DS-8608N-I8/MB-84
DS-8616N-I16(16*6T)
DS-8664N-I16(16*6T)
DS-GJZ9N08-F96(B)
DS-FN64CGX(B)
DS-7732N-I4
DS-7708N-I4
DS-7716N-I4/16P
DS-7732N-I4/16P

Product number
DS-7708N-I4/8P
DS-7732N-I4/24P
DS-7716N-I4
DS-7608N-I2
DS-7616N-I2
DS-7708N-I4/8P/1G
DS-7616N-I2/16P
DS-7608N-I2/8P
DS-7632N-I2/16P
DS-7632N-I2

B.8 Country and region codes

The following is a list of country and region codes supported by the license plate recognition algorithm library.

CRIndex (country and region code)	Countries and regions
0	Algorithm not supported
1	Czech Republic
2	France
3	Germany
4	Spain
5	Italy
6	Netherlands
7	Poland
8	Slovakia

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
9	Belarus
10	Moldova
11	Russia
12	Ukraine
13	Belgium
14	Bulgaria
15	Denmark
16	Finland
17	U.K.
18	Greece
19	Croatia
20	Hungary
21	Israel(Asia)
22	Luxembourg
23	Republic of Macedonia (renamed the Republic of North Macedonia in 2018)
24	Norway
25	Portugal
26	Romania
27	Serbia
28	Azerbaijan Republic (Asia)
29	Georgia (Asia)
30	Kazakhstan (Asia)
31	Lithuania

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
32	Turkmenistan (Asia)
33	Uzbekistan(Asia)
34	Latvia
35	Estonia
36	Albania
37	Austria
38	Bosnia and Herzegovina
39	Ireland
40	Iceland
41	Vatican City
42	malta
43	Sweden
44	Switzerland
45	Cyprus
46	Türkiye
47	slovenia
48	Montenegro
49	Kosovo
50	Andorra
51	Armenia (Asia)
52	Monaco
53	Liechtenstein
54	san marino

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
55	reserved text
56	reserved text
57	reserved text
58	reserved text
59	China
60	Bahrain
61	South Korea
62	Lebanon
63	Nepal
64	Thailand
65	Pakistan
66	United Arab Emirates
67	Bhutan
68	Oman
69	North Korea
70	the Philippines
71	Cambodia
72	Qatar
73	Kyrgyzstan
74	Maldives
75	Malaysia
76	Mongolia
77	Saudi Arabia
78	Brunei

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
79	Laos
80	Japan
81	Türkiye
82	State of Palestine
83	Tajikistan
84	Kuwait
85	Syria
86	India
87	Indonesia
88	Afghanistan
89	Sri Lanka
90	Iraq
91	Vietnam
92	Iran
93	Yemen
94	Jordan
95	Myanmar
96	Sikkim
97	Bangladesh
98	Singapore
99	East Timor
100	reserved text
101	reserved text
102	reserved text

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
103	reserved text
104	Egypt
105	Libya
106	Sudan
107	Tunisia
108	Algeria
109	Morocco
110	Ethiopia
111	Eritrea
112	somalia
113	Djibouti
114	Kenya
115	Tanzania
116	Uganda
117	Rwanda
118	Burundi
119	Seychelles
120	Chad
121	Central Africa
122	Cameroon
123	Equatorial Guinea
124	Gabon
125	Republic of the Congo (ie: Congo (Brazzaville))
126	Democratic Republic of the Congo (DRC)

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
127	Sao Tome and Principe
128	mauritania
129	Western Sahara (Sahrawi)
130	Senegal
131	Gambia
132	Mali
133	Burkina Faso
134	Guinea
135	Guinea-Bissau
136	cape verde
137	Sierra Leone
138	Liberia
139	Ivory Coast
140	Ghana
141	Togo
142	Benin
143	Niger
144	Zambia
145	Angola
146	Zimbabwe
147	Malawi
148	Mozambique
149	Botswana
150	Namibia

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
151	South Africa
152	Swaziland
153	Lesotho
154	Madagascar
155	Comoros
156	Mauritius
157	Nigeria
158	South Sudan
159	Saint Helena (British Overseas Territory)
160	Mayotte (French overseas department)
161	Reunion (French overseas department)
162	Canary Islands (Spanish overseas territory)
163	Azores (North Atlantic Ocean, west coast of Africa (Portugal))
164	Madeira Islands (North Atlantic Ocean, west coast of Africa (Portugal))
165	reserved text
166	reserved text
167	reserved text
168	reserved text
169	Canada
170	Greenland (an autonomous overseas territory of the Kingdom of Denmark)
171	Saint-Pierre and Miquelon (French overseas territory)
172	USA
173	Bermuda (British Overseas Territory)

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
174	Mexico
175	Guatemala
176	Belize
177	El Salvador
178	Honduras
179	Nicaragua
180	costa rica
181	Panama
182	Bahamas
183	Turks and Caicos Islands (British Overseas Territory)
184	Cuba
185	Jamaica
186	Cayman Islands (British Overseas Territory)
187	Haiti
188	dominican
189	Puerto Rico (U.S. overseas territory)
190	United States Virgin Islands
191	The British Virgin Islands
192	Anguilla (British Overseas Territory)
193	Antigua and Barbuda
194	Saint Martin (French overseas territory)
195	Sint Maarten (autonomous country of the Kingdom of the Netherlands)
196	Saint Barthelemy (French overseas territory)

CRIndex (country and region code)	Countries and regions
197	Saint Kitts and Nevis
198	Montserrat (British Overseas Territory)
199	Guadeloupe (French overseas territory)
200	Dominica
201	Martinique (French overseas territory)
202	saint lucia
203	Saint Vincent and the Grenadines
204	Grenada
205	Barbados
206	Trinidad and Tobago
207	Curaçao (an autonomous country within the Kingdom of the Netherlands)
208	Aruba (an autonomous country within the Kingdom of the Netherlands)
209	Netherlands Antilles
210	Colombia
211	Venezuela
212	Guyana
213	Surinam
214	French Guiana
215	Ecuador
216	Peru
217	Bolivia
218	Paraguay
219	Chile

CRIndex (country and region code)	Countries and regions
220	Brazil
221	Uruguay
222	Argentina
223	reserved text
224	reserved text
225	reserved text
226	reserved text
227	Australia
228	New Zealand
229	papua new guinea
230	solomon islands
231	Vanuatu
232	New Caledonia (French overseas territory)
233	Palau
234	Federated States of Micronesia
235	Marshall Islands
236	Northern Mariana Islands (U.S. overseas territory)
237	Guam (U.S. overseas territory)
238	Nauru
239	Kiribati
240	Fiji Islands
241	Tonga
242	Tuvalu

ISUPSDK (General) Development Guide

CRIndex (country and region code)	Countries and regions
243	Wallis and Futuna (French overseas territory)
244	Samoa
245	American Samoa
246	Tokelau (New Zealand)
247	Niue (New Zealand)
248	Cook Islands (New Zealand)
249	French Polynesia
250	Pitcairn Islands (British Overseas Territory)
251	Hawaii (a U.S. overseas territory)
252	reserved text
253	reserved text
254	Unrecognizable
255	all
256	Taiwan, China
257	China Hong Kong
258	Macao, China

B.9 Area Code

The following is a list of area codes supported by the license plate recognition algorithm library.

region (region code)	area
IS	Russian speaking area
EU	Europe
EUandCIS	Europe and Russia

ISUPSDK (General) Development Guide

region (region code)	area
ME	middle East
other	other
APAC	Asia-Pacific
AFandAM	Africa and America
THAandLA	Thailand Laos
HKandMO	Hong Kong and Macau
India	India
All	All regions

