



# **Device Network SDK (Person-Based Access Control)**

**Developer Guide**

## Legal Information

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE DOCUMENT IS PROVIDED "AS IS" AND "WITH ALL FAULTS AND ERRORS". OUR COMPANY MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IN NO EVENT WILL OUR COMPANY BE LIABLE FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES, INCLUDING, AMONG OTHERS, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF DATA, CORRUPTION OF SYSTEMS, OR LOSS OF DOCUMENTATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, IN CONNECTION WITH THE USE OF THE DOCUMENT, EVEN IF OUR COMPANY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR LOSS.

# Contents

<b>Chapter 1 Access Control .....</b>	<b>1</b>
1.1 Introduction .....	1
1.2 Update History .....	1
<b>Chapter 2 Typical Applications .....</b>	<b>23</b>
2.1 Data Collection .....	23
2.1.1 Online Collect Data .....	23
2.1.2 Offline Collect Data .....	26
2.2 Manage Person Information .....	31
2.3 Manage Card Information .....	35
2.3.1 Collect Card Information .....	38
2.3.2 Card Operation .....	39
2.4 Manage Fingerprint Information .....	41
2.4.1 Fingerprint Collection .....	45
2.5 Manage Face Information .....	46
2.5.1 Create Face Picture Library .....	46
2.5.2 Collect Face Data .....	48
2.5.3 Manage Face Records in Face Picture Library .....	49
2.5.4 Configure Facial Recognition Mode .....	52
2.5.5 Other Facial Applications .....	54
2.6 Configure Anti-Passing Back .....	56
2.7 Cross-Controller Anti-Passing Back Configuration .....	59
2.7.1 Configure Route Anti-Passing Back Based on Network .....	59
2.7.2 Configure Entrance/Exit Anti-Passing Back Based on Network .....	63
2.7.3 Configure Route Anti-Passing Back Based on Card .....	66
2.7.4 Configure Entrance/Exit Anti-Passing Back Based on Card .....	69
2.8 Schedule Settings .....	71

2.8.1 Configure Access Permission Control Schedule .....	71
2.8.2 Configure Access Permission Control Schedule (Integrate by Transmitting Text Protocol) .....	77
2.8.3 Configure Authentication Mode Control Schedule .....	79
2.8.4 Configure Door Control Schedule .....	86
2.9 Alarm and Event Receiving .....	92
2.9.1 Access Control Event Types .....	93
2.9.2 Configure Access Control Event .....	110
2.9.3 Supported Alarm/Event Types and Details .....	112
2.9.4 Configure Mask Detection Event .....	113
2.9.5 Configure Hard Hat Detection Event .....	114
2.9.6 Receive Alarm/Event in Arming Mode .....	115
2.9.7 Receive Alarm/Event in Listening Mode .....	118
2.9.8 Search for Access Control Events .....	121
2.9.9 Remotely Verify Access Control Events .....	125
2.10 Remotely Control Door, Elevator, and Buzzer .....	125
2.11 Configure Attendance Status and Schedule .....	126
2.12 Turnstile Settings .....	131
2.13 Other Applications .....	134
2.13.1 Device Settings .....	134
2.13.2 Multi-Factor Authentication .....	137
2.13.3 Temperature Measurement .....	138
2.13.4 Other Configurations .....	139
2.14 Integrate by Transmitting Text Protocol .....	144
<b>Chapter 3 API Reference .....</b>	<b>146</b>
3.1 NET_DVR_Cleanup .....	146
3.2 NET_DVR_CloseAlarmChan_V30 .....	146
3.3 NET_DVR_ControlGateway .....	147

3.4 NET_DVR_GetDeviceAbility .....	147
3.5 NET_DVR_GetDeviceConfig .....	148
3.6 NET_DVR_GetDownloadState .....	149
3.7 NET_DVR_GetDVRConfig .....	150
3.8 NET_DVR_GetErrorMsg .....	151
3.9 NET_DVR_GetLastError .....	152
3.10 NET_DVR_GetNextRemoteConfig .....	152
3.11 NET_DVR_GetSDKLocalCfg .....	153
3.12 NET_DVR_GetUploadState .....	154
3.13 NET_DVR_Init .....	156
3.14 NET_DVR_Login_V40 .....	156
3.15 NET_DVR_Logout .....	157
3.16 NET_DVR_SDKChannelToISAPI .....	157
3.17 NET_DVR_SendRemoteConfig .....	158
3.18 NET_DVR_SetConnectTime .....	159
3.19 NET_DVR_SetDeviceConfig .....	160
3.20 NET_DVR_SetDVRConfig .....	161
3.21 NET_DVR_SetDVRMessageCallBack_V50 .....	162
3.22 NET_DVR_SetSDKInitCfg .....	170
3.23 NET_DVR_SetSDKLocalCfg .....	171
3.24 NET_DVR_SetupAlarmChan_V50 .....	172
3.25 NET_DVR_StartDownload .....	172
3.26 NET_DVR_StartListen_V30 .....	173
3.27 NET_DVR_StartRemoteConfig .....	174
3.28 NET_DVR_STDXMLConfig .....	175
3.29 NET_DVR_StopDownload .....	177
3.30 NET_DVR_StopListen_V30 .....	177
3.31 NET_DVR_StopRemoteConfig .....	177

3.32 NET_DVR_UploadClose .....	178
3.33 NET_DVR_UploadFile_V40 .....	178
3.34 Callback Function .....	179
3.34.1 fLoginResultCallBack .....	179
3.34.2 fRemoteConfigCallback .....	180
3.34.3 MSGCallBack .....	181
<b>Appendix A. Data Structure .....</b>	<b>183</b>
A.1 CHAR_ENCODE_CONVERT .....	183
A.2 NET_ALARM_RECORD_EXCEPTION .....	184
A.3 NET_ALARM_STREAM_EXCEPTION .....	184
A.4 NET_ALARM_RESOURCE_USAGE .....	184
A.5 NET_ALARM_RECORDFILE LOSS .....	185
A.6 NET_ALARM_CVR_SUBINFO_UNION .....	185
A.7 NET_DVR_ACS_ALARM_INFO .....	186
A.8 NET_DVR_ACS_CFG .....	188
A.9 NET_DVR_ACS_EVENT_CFG .....	189
A.10 NET_DVR_ACS_EVENT_COND .....	191
A.11 NET_DVR_ACS_EVENT_DETAIL .....	193
A.12 NET_DVR_ACS_EVENT_INFO .....	197
A.13 NET_DVR_ACS_EVENT_INFO_EXTEND .....	200
A.14 NET_DVR_ACS_EVENT_INFO_EXTEND_V20 .....	202
A.15 NET_DVR_ACS_WORK_STATUS_V50 .....	204
A.16 NET_DVR_ALARMER .....	206
A.17 NET_DVR_ALARMINFO_DEV .....	207
A.18 NET_DVR_ALARMINFO_DEV_V40 .....	208
A.19 NET_DVR_ALARMINFO_V30 .....	209
A.20 NET_DVR_ALARMINFO_V40 .....	210
A.21 NET_DVR_ALRAM_FIXED_HEADER .....	211

A.22 NET_DVR_ALARM_ISAPI_INFO .....	215
A.23 NET_DVR_ALARM_ISAPI_PICDATA .....	216
A.24 NET_DVR_CAPTURE_FACE_CFG .....	216
A.25 NET_DVR_CAPTURE_FACE_COND .....	218
A.26 NET_DVR_CAPTURE_FINGERPRINT_CFG .....	218
A.27 NET_DVR_CAPTURE_FINGERPRINT_COND .....	219
A.28 NET_DVR_CARD_READER_CFG_V50 .....	220
A.29 NET_DVR_CARD_READER_PLAN .....	224
A.30 NET_DVR_CETTIFICATE_INFO .....	225
A.31 NET_DVR_CHECK_FACE_PICTURE_CFG .....	226
A.32 NET_DVR_CHECK_FACE_PICTURE_COND .....	227
A.33 NET_DVR_CHECK_FACE_PICTURE_STATUS .....	227
A.34 NET_DVR_DATE .....	228
A.35 NET_DVR_DEL_FINGER_PRINT_MODE_V50 .....	228
A.36 NET_DVR_DEVICEINFO_V30 .....	229
A.37 NET_DVR_DEVICEINFO_V40 .....	233
A.38 NET_DVR_DOOR_CFG .....	236
A.39 NET_DVR_DOOR_FILE_UPLOAD_PARAM .....	238
A.40 NET_DVR_DOOR_STATUS_PLAN .....	239
A.41 NET_DVR_ETHERNET_V30 .....	239
A.42 NET_DVR_EVENT_CARD_LINKAGE_CFG_V51 .....	240
A.43 NET_DVR_EVENT_CARD_LINKAGE_COND .....	242
A.44 NET_DVR_EVENT_LINKAGE_INFO .....	243
A.45 NET_DVR_EVETN_CARD_LINKAGE_UNION .....	243
A.46 NET_DVR_FACE_FEATURE .....	244
A.47 NET_DVR_FAILED_FACE_COND .....	245
A.48 NET_DVR_FAILED_FACE_INFO .....	245
A.49 NET_DVR_FINGER_PRINT_BYCARD_V50 .....	246

A.50 NET_DVR_FINGER_PRINT_BYREADER_V50 .....	247
A.51 NET_DVR_FINGER_PRINT_CFG_V50 .....	248
A.52 NET_DVR_FINGER_PRINT_INFO_COND_V50 .....	249
A.53 NET_DVR_FINGER_PRINT_INFO_CTRL_V50 .....	250
A.54 NET_DVR_FINGER_PRINT_STATUS_V50 .....	251
A.55 NET_DVR_GROUP_CFG .....	252
A.56 NET_DVR_GROUP_COMBINATION_INFO_V50 .....	253
A.57 NET_DVR_HOLIDAY_GROUP_CFG .....	254
A.58 NET_DVR_HOLIDAY_PLAN_CFG .....	254
A.59 NET_DVR_ID_CARD_INFO .....	255
A.60 NET_DVR_ID_CARD_INFO_ALARM .....	257
A.61 NET_DVR_ID_CARD_INFO_EXTEND .....	259
A.62 NET_DVR_INIT_CFG_ABILITY .....	260
A.63 NET_DVR_IPADDR_UNION .....	261
A.64 NET_DVR_JSON_DATA_CFG .....	261
A.65 NET_DVR_LOCAL_ABILITY_PARSE_CFG .....	262
A.66 NET_DVR_LOCAL_ASYNC_CFG .....	263
A.67 NET_DVR_LOCAL_BYTE_ENCODE_CONVERT .....	264
A.68 NET_DVR_LOCAL_CERTIFICATION .....	264
A.69 NET_DVR_LOCAL_CFG_TYPE_PTZ .....	265
A.70 NET_DVR_LOCAL_CHECK_DEV .....	266
A.71 NET_DVR_LOCAL_GENERAL_CFG .....	266
A.72 NET_DVR_LOCAL_LOG_CFG .....	267
A.73 NET_DVR_LOCAL_MEM_POOL_CFG .....	268
A.74 NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG .....	268
A.75 NET_DVR_LOCAL_PORT_MULTI_CFG .....	269
A.76 NET_DVR_LOCAL_PROTECT_KEY_CFG .....	270
A.77 NET_DVR_LOCAL_SDK_PATH .....	270

A.78 NET_DVR_LOCAL_STREAM_CALLBACK_CFG .....	270
A.79 NET_DVR_LOCAL_TALK_MODE_CFG .....	271
A.80 NET_DVR_LOCAL_TCP_PORT_BIND_CFG .....	271
A.81 NET_DVR_LOCAL_UDP_PORT_BIND_CFG .....	272
A.82 NET_DVR_MESSAGE_CALLBACK_PARAM_V51 .....	273
A.83 NET_DVR_MIME_UNIT .....	273
A.84 NET_DVR_MULTI_CARD_CFG_V50 .....	274
A.85 NET_DVR_MULTI_CARD_GROUP_CFG_V50 .....	275
A.86 NET_DVR_NETCFG_V50 .....	275
A.87 NET_DVR_PLAN_TEMPLATE .....	277
A.88 NET_DVR_PPPOECFG .....	278
A.89 NET_DVR_RECORD_PASSBACK_MANUAL_COND .....	278
A.90 NET_DVR_RECORD_PASSBACK_MANUAL_TASK_RET .....	279
A.91 NET_DVR_RTSP_PARAMS_CFG .....	280
A.92 NET_DVR_SETUPALARM_PARAM_V50 .....	280
A.93 NET_DVR_SIMPLE_DAYTIME .....	283
A.94 NET_DVR_SIMXML_LOGIN .....	284
A.95 NET_DVR_SINGLE_PLAN_SEGMENT .....	284
A.96 NET_DVR_STREAM_INFO .....	285
A.97 NET_DVR_TIME .....	286
A.98 NET_DVR_TIME_EX .....	286
A.99 NET_DVR_TIME_SEGMENT .....	286
A.100 NET_DVR_TIME_V30 .....	287
A.101 NET_DVR_USER_LOGIN_INFO .....	288
A.102 NET_DVR_VALID_PERIOD_CFG .....	289
A.103 NET_DVR_WEEK_PLAN_CFG .....	290
A.104 NET_DVR_XML_CONFIG_INPUT .....	291
A.105 NET_DVR_XML_CONFIG_OUTPUT .....	292

A.106 NET_SDK_CALLBACK_STATUS_NORMAL .....	292
A.107 NET_SDK_DOWNLOAD_TYPE .....	293
A.108 NET_SDK_LOCAL_CFG_TYPE .....	297
A.109 NET_SDK_UPLOAD_TYPE .....	299
A.110 NET_VCA_POINT .....	302
A.111 NET_VCA_RECT .....	303
<b>Appendix B. Event Linkage Types .....</b>	<b>304</b>
<b>Appendix C. HCNetSDK Log Types .....</b>	<b>316</b>
<b>Appendix D. Device Network SDK Errors .....</b>	<b>352</b>
<b>Appendix E. Request URLs .....</b>	<b>396</b>
E.1 /ISAPI/AccessControl/AcsCfg/capabilities?format=json .....	398
E.2 /ISAPI/AccessControl/AcsCfg?format=json .....	398
E.3 /ISAPI/AccessControl/AcsEvent/capabilities?format=json .....	399
E.4 /ISAPI/AccessControl/AcsEvent/StorageCfg/capabilities?format=json .....	400
E.5 /ISAPI/AccessControl/AcsEvent/StorageCfg?format=json .....	400
E.6 /ISAPI/AccessControl/AcsEvent?format=json .....	401
E.7 /ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json .....	402
E.8 /ISAPI/AccessControl/AcsEventTotalNum?format=json .....	403
E.9 /ISAPI/AccessControl/AntiSneakCfg/capabilities?format=json .....	404
E.10 /ISAPI/AccessControl/AntiSneakCfg?format=json .....	404
E.11 /ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json .....	405
E.12 /ISAPI/AccessControl/Attendance/planTemplate/capabilities?format=json .....	406
E.13 /ISAPI/AccessControl/Attendance/planTemplate?format=json .....	406
E.14 /ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json .....	406
E.15 /ISAPI/AccessControl/Attendance/weekPlan/capabilities?format=json .....	407
E.16 /ISAPI/AccessControl/blackObject/capabilities?format=json .....	408
E.17 /ISAPI/AccessControl/blackObject?format=json .....	408
E.18 /ISAPI/AccessControl/capabilities .....	409

E.19 /ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json .....	409
E.20 /ISAPI/AccessControl/CaptureCardInfo?format=json .....	410
E.21 /ISAPI/AccessControl/CaptureFingerPrint .....	410
E.22 /ISAPI/AccessControl/CaptureFingerPrint/capabilities .....	411
E.23 /ISAPI/AccessControl/CaptureIDInfo/capabilities?format=json .....	411
E.24 /ISAPI/AccessControl/CaptureIDInfo?format=json .....	411
E.25 /ISAPI/AccessControl/CapturePresetParam/capabilities?format=json .....	412
E.26 /ISAPI/AccessControl/CapturePresetParam?format=json .....	413
E.27 /ISAPI/AccessControl/CaptureRule/capabilities?format=json .....	414
E.28 /ISAPI/AccessControl/CaptureRule?format=json .....	414
E.29 /ISAPI/AccessControl/CardInfo/capabilities?format=json .....	415
E.30 /ISAPI/AccessControl/CardInfo/Count?format=json .....	415
E.31 /ISAPI/AccessControl/CardInfo/Count?format=json&employeeNo=<ID> .....	416
E.32 /ISAPI/AccessControl/CardInfo/Delete?format=json .....	416
E.33 /ISAPI/AccessControl/CardInfo/Modify?format=json .....	417
E.34 /ISAPI/AccessControl/CardInfo/Record?format=json .....	417
E.35 /ISAPI/AccessControl/CardInfo/Search?format=json .....	418
E.36 /ISAPI/AccessControl/CardInfo/SetUp?format=json .....	418
E.37 /ISAPI/AccessControl/CardOperations/capabilities?format=json .....	419
E.38 /ISAPI/AccessControl/CardOperations/cardParam?format=json .....	420
E.39 /ISAPI/AccessControl/CardOperations/clearData?format=json .....	420
E.40 /ISAPI/AccessControl/CardOperations/controlBlock?format=json .....	420
E.41 /ISAPI/AccessControl/CardOperations/customData/searchTask?format=json .....	421
E.42 /ISAPI/AccessControl/CardOperations/customData?format=json .....	422
E.43 /ISAPI/AccessControl/CardOperations/dataBlock/<address>?format=json .....	422
E.44 /ISAPI/AccessControl/CardOperations/dataBlock/control?format=json .....	423
E.45 /ISAPI/AccessControl/CardOperations/dataTrans?format=json .....	423
E.46 /ISAPI/AccessControl/CardOperations/encryption?format=json .....	424

E.47 /ISAPI/AccessControl/CardOperations/protocol?format=json .....	424
E.48 /ISAPI/AccessControl/CardOperations/reset?format=json .....	424
E.49 /ISAPI/AccessControl/CardOperations/sectionEncryption?format=json .....	425
E.50 /ISAPI/AccessControl/CardOperations/verification?format=json .....	426
E.51 /ISAPI/AccessControl/CardReaderAntiSneakCfg/capabilities?format=json .....	426
E.52 /ISAPI/AccessControl/CardReaderAntiSneakCfg/<ID>?format=json .....	427
E.53 /ISAPI/AccessControl/CardReaderCfg/<ID>?format=json .....	427
E.54 /ISAPI/AccessControl/CardReaderCfg/capabilities?format=json .....	428
E.55 /ISAPI/AccessControl/CardVerificationRule/capabilities?format=json .....	429
E.56 /ISAPI/AccessControl/CardVerificationRule/progress?format=json .....	429
E.57 /ISAPI/AccessControl/CardVerificationRule?format=json .....	430
E.58 /ISAPI/AccessControl/ChannelControllerCfg .....	430
E.59 /ISAPI/AccessControl/ChannelControllerCfg/capabilities .....	431
E.60 /ISAPI/AccessControl/channelControllerTypeCfg/capabilities?format=json .....	431
E.61 /ISAPI/AccessControl/channelControllerTypeCfg?format=json .....	432
E.62 /ISAPI/AccessControl/ClearAntiSneakCfg/capabilities?format=json .....	432
E.63 /ISAPI/AccessControl/ClearAntiSneakCfg?format=json .....	433
E.64 /ISAPI/AccessControl/ClearAntiSneak?format=json .....	433
E.65 /ISAPI/AccessControl/ClearAntiSneak/capabilities?format=json .....	434
E.66 /ISAPI/AccessControl/ClearAttendancePlan?format=json .....	434
E.67 /ISAPI/AccessControl/ClearCardRecord .....	434
E.68 /ISAPI/AccessControl/ClearCardRecord/capabilities .....	435
E.69 /ISAPI/AccessControl/ClearPictureCfg/capabilities?format=json .....	435
E.70 /ISAPI/AccessControl/ClearPictureCfg?format=json .....	436
E.71 /ISAPI/AccessControl/ClearPlansCfg/capabilities?format=json .....	436
E.72 /ISAPI/AccessControl/ClearPlansCfg?format=json .....	437
E.73 /ISAPI/AccessControl/ClearSubmarineBack .....	437
E.74 /ISAPI/AccessControl/ClearSubmarineBack/capabilities .....	438

E.75 /ISAPI/AccessControl/Configuration/attendanceMode/capabilities?format=json .....	438
E.76 /ISAPI/AccessControl/Configuration/attendanceMode?format=json .....	439
E.77 /ISAPI/AccessControl/Configuration/IRCfg/capabilities?format=json .....	439
E.78 /ISAPI/AccessControl/Configuration/IRCfg?format=json .....	440
E.79 /ISAPI/AccessControl/Configuration/NFCCfg/capabilities?format=json .....	441
E.80 /ISAPI/AccessControl/Configuration/NFCCfg?format=json .....	441
E.81 /ISAPI/AccessControl/Configuration/RFCardCfg/capabilities?format=json .....	442
E.82 /ISAPI/AccessControl/Configuration/RFCardCfg?format=json .....	442
E.83 /ISAPI/AccessControl/Configuration/safetyHelmetDetection/capabilities?format=json .	443
E.84 /ISAPI/AccessControl/Configuration/safetyHelmetDetection?format=json .....	444
E.85 /ISAPI/AccessControl/DeployInfo .....	444
E.86 /ISAPI/AccessControl/DeployInfo/capabilities .....	445
E.87 /ISAPI/AccessControl/EventOptimizationCfg/capabilities?format=json .....	445
E.88 /ISAPI/AccessControl/EventOptimizationCfg?format=json .....	446
E.89 /ISAPI/AccessControl/FaceCompareCond .....	446
E.90 /ISAPI/AccessControl/FaceCompareCond/capabilities .....	447
E.91 /ISAPI/AccessControl/FaceRecognizeMode/capabilities?format=json .....	448
E.92 /ISAPI/AccessControl/FaceRecognizeMode?format=json .....	448
E.93 /ISAPI/AccessControl/FaceTemperatureEvent/capabilities?format=json .....	449
E.94 /ISAPI/AccessControl/FaceTemperatureEvent?format=json .....	449
E.95 /ISAPI/AccessControl/FingerPrint/DeleteProcess?format=json .....	450
E.96 /ISAPI/AccessControl/FingerPrint/Delete/capabilities?format=json .....	450
E.97 /ISAPI/AccessControl/FingerPrint/Delete?format=json .....	451
E.98 /ISAPI/AccessControl/FingerPrint/SetUp?format=json .....	451
E.99 /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json .....	452
E.100 /ISAPI/AccessControl/FingerPrintDownload?format=json .....	453
E.101 /ISAPI/AccessControl/FingerPrintModify?format=json .....	453
E.102 /ISAPI/AccessControl/FingerPrintProgress?format=json .....	454

E.103 /ISAPI/AccessControl/FingerPrintUpload?format=json .....	454
E.104 /ISAPI/AccessControl/GetAcsEvent/capabilities .....	455
E.105 /ISAPI/AccessControl/healthCodeCfg/capabilities?format=json .....	456
E.106 /ISAPI/AccessControl/healthCodeCfg?format=json .....	456
E.107 /ISAPI/AccessControl/IDCardInfoEvent/capabilities?format=json .....	457
E.108 /ISAPI/AccessControl/IDCardInfoEvent?format=json .....	457
E.109 /ISAPI/AccessControl/IdentityTerminal .....	458
E.110 /ISAPI/AccessControl/IdentityTerminal/capabilities .....	458
E.111 /ISAPI/AccessControl/keyCfg/<ID>/attendance?format=json .....	459
E.112 /ISAPI/AccessControl/keyCfg/attendance/capabilities?format=json .....	460
E.113 /ISAPI/AccessControl/keyCfg/attendance?format=json .....	460
E.114 /ISAPI/AccessControl/LogModeCfg/capabilities?format=json .....	460
E.115 /ISAPI/AccessControl/LogModeCfg?format=json .....	461
E.116 /ISAPI/AccessControl/maskDetection/capabilities?format=json .....	462
E.117 /ISAPI/AccessControl/maskDetection?format=json .....	462
E.118 /ISAPI/AccessControl/OfflineCapture/capabilities?format=json .....	463
E.119 /ISAPI/AccessControl/OfflineCapture/DataCollections/<captureNo>?format=json .....	463
E.120 /ISAPI/AccessControl/OfflineCapture/DataCollections/searchTask?format=json .....	464
E.121 /ISAPI/AccessControl/OfflineCapture/DataCollections?format=json .....	464
E.122 /ISAPI/AccessControl/OfflineCapture/dataOutput/progress?format=json .....	464
E.123 /ISAPI/AccessControl/OfflineCapture/dataOutput?format=json .....	465
E.124 /ISAPI/AccessControl/OfflineCapture/progress?format=json .....	465
E.125 /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json .....	466
E.126 /ISAPI/AccessControl/OfflineCapture/uploadFailedDetails?format=json .....	467
E.127 /ISAPI/AccessControl/OSDPMModify/<ID>?format=json .....	467
E.128 /ISAPI/AccessControl/OSDPMModify/capabilities?format=json .....	467
E.129 /ISAPI/AccessControl/OSDPStatus/<ID>?format=json .....	468
E.130 /ISAPI/AccessControl/OSDPStatus/capabilities?format=json .....	468

E.131 /ISAPI/AccessControl/personInfoExtendName/capabilities?format=json .....	469
E.132 /ISAPI/AccessControl/personInfoExtendName?format=json .....	469
E.133 /ISAPI/AccessControl/QRCodeEvent/capabilities?format=json .....	470
E.134 /ISAPI/AccessControl/QRCodeEvent?format=json .....	471
E.135 /ISAPI/AccessControl/ReaderAcrossHost .....	471
E.136 /ISAPI/AccessControl/ReaderAcrossHost/capabilities .....	472
E.137 /ISAPI/AccessControl/remoteCheck/capabilities?format=json .....	472
E.138 /ISAPI/AccessControl/remoteCheck?format=json .....	473
E.139 /ISAPI/AccessControl/RemoteControl/buzzer/<ID>?format=json .....	473
E.140 /ISAPI/AccessControl/RemoteControl/buzzer/capabilities?format=json .....	473
E.141 /ISAPI/AccessControl/remoteCtrlleModeCfg/capabilities?format=json .....	474
E.142 /ISAPI/AccessControl/remoteCtrlleModeCfg?format=json .....	474
E.143 /ISAPI/AccessControl/ServerDevice .....	475
E.144 /ISAPI/AccessControl/ServerDevice/capabilities .....	476
E.145 /ISAPI/AccessControl/showHealthCodeCfg/capabilities?format=json .....	476
E.146 /ISAPI/AccessControl/showHealthCodeCfg?format=json .....	477
E.147 /ISAPI/AccessControl/SubmarineBackMode .....	477
E.148 /ISAPI/AccessControl/SubmarineBackMode/capabilities .....	478
E.149 /ISAPI/AccessControl/SubmarineBackReader/capabilities .....	479
E.150 /ISAPI/AccessControl/SubmarineBackReader/ConfigureNo/<ID> .....	479
E.151 /ISAPI/AccessControl/StartReaderInfo .....	480
E.152 /ISAPI/AccessControl/StartReaderInfo/capabilities .....	480
E.153 /ISAPI/AccessControl/SubmarineBackHostInfo/capabilities .....	481
E.154 /ISAPI/AccessControl/SubmarineBack/capabilities .....	481
E.155 /ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID> .....	482
E.156 /ISAPI/AccessControl/SubmarineBack .....	483
E.157 /ISAPI/AccessControl/temperatureMeasureAreaCalibration/capabilities?format=json .....	483

E.158 /ISAPI/AccessControl/temperatureMeasureAreaCalibration?format=json .....	484
E.159 /ISAPI/AccessControl/temperatureMeasureAreaCfg/capabilities?format=json .....	485
E.160 /ISAPI/AccessControl/temperatureMeasureAreaCfg?format=json .....	485
E.161 /ISAPI/AccessControl/temperatureMeasureCfg/capabilities?format=json .....	486
E.162 /ISAPI/AccessControl/temperatureMeasureCfg?format=json .....	486
E.163 /ISAPI/AccessControl/UserInfo/capabilities?format=json .....	487
E.164 /ISAPI/AccessControl/UserInfo/Count?format=json .....	488
E.165 /ISAPI/AccessControl/UserInfo/Delete?format=json .....	488
E.166 /ISAPI/AccessControl/UserInfo/Modify?format=json .....	489
E.167 /ISAPI/AccessControl/UserInfo/Record?format=json .....	489
E.168 /ISAPI/AccessControl/UserInfo/Search?format=json .....	489
E.169 /ISAPI/AccessControl/UserInfo/SetUp?format=json .....	490
E.170 /ISAPI/AccessControl/UserInfoDetail/Delete/capabilities?format=json .....	491
E.171 /ISAPI/AccessControl/UserInfoDetail/Delete?format=json .....	492
E.172 /ISAPI/AccessControl/UserInfoDetail/DeleteProcess?format=json .....	492
E.173 /ISAPI/AccessControl/UserRightPlanTemplate/capabilities?format=json .....	493
E.174 /ISAPI/AccessControl/UserRightPlanTemplate/<TemplateNo>?format=json .....	494
E.175 /ISAPI/AccessControl/UserRightHolidayGroupCfg/<GroupNo>?format=json .....	495
E.176 /ISAPI/AccessControl/UserRightHolidayGroupCfg/capabilities?format=json .....	496
E.177 /ISAPI/AccessControl/UserRightHolidayPlanCfg/<PlanNo>?format=json .....	496
E.178 /ISAPI/AccessControl/UserRightHolidayPlanCfg/capabilities?format=json .....	498
E.179 /ISAPI/AccessControl/UserRightWeekPlanCfg/<PlanNo>?format=json .....	498
E.180 /ISAPI/AccessControl/UserRightWeekPlanCfg/capabilities?format=json .....	499
E.181 /ISAPI/Intelligent/FDLib/capabilities?format=json .....	500
E.182 /ISAPI/Intelligent/FDLib?format=json .....	501
E.183 /ISAPI/Intelligent/FDLib?format=json&FDID=&faceLibType= .....	502
E.184 /ISAPI/Intelligent/FDLib/Count?format=json .....	504
E.185 /ISAPI/Intelligent/FDLib/Count?format=json&FDID=&faceLibType= .....	504

E.186 /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json .....	505
E.187 /ISAPI/Intelligent/FDLib/FDSearch?format=json .....	506
E.188 /ISAPI/Intelligent/FDLib/FDModify?format=json .....	506
E.189 /ISAPI/Intelligent/FDLib/FDSearch/Delete?format=json&FDID=&faceLibType= .....	507
E.190 /ISAPI/Intelligent/FDLib/FDSetUp?format=json .....	507
E.191 /ISAPI/System/capabilities .....	508
E.192 /ISAPI/System/PictureServer?format=json .....	509
<b>Appendix F. Request and Response Messages .....</b>	<b>510</b>
F.1 JSON_AcsCfg .....	510
F.2 JSON_AcsEventCond .....	512
F.3 JSON_AcsEvent .....	513
F.4 JSON_AcsEventTotalNum .....	517
F.5 JSON_AcsEventTotalNumCond .....	518
F.6 JSON_AddFaceRecordCond .....	518
F.7 JSON_AddFaceRecordResult .....	522
F.8 JSON_AntiSneakCfg .....	522
F.9 JSON_Attendance .....	522
F.10 JSON_AttendanceCap .....	523
F.11 JSON_AttendanceList .....	523
F.12 JSON_AttendanceMode .....	524
F.13 JSON_AttendancePlanTemplate .....	524
F.14 JSON_AttendancePlanTemplateCap .....	524
F.15 JSON_AttendancePlanTemplateList .....	525
F.16 JSON_AttendanceWeekPlan .....	526
F.17 JSON_AttendanceWeekPlanCap .....	526
F.18 JSON_BatchEditFaceRecord .....	527
F.19 JSON_BlackBodyCfg .....	529
F.20 JSON_Cap_AcsCfg .....	529

F.21 JSON_Cap_AcsEvent .....	532
F.22 JSON_Cap_AcsEventTotalNum .....	540
F.23 JSON_Cap_AntiSneakCfg .....	542
F.24 JSON_Cap_AttendanceMode .....	543
F.25 JSON_Cap_BlackBodyCfg .....	543
F.26 JSON_Cap_CardInfo .....	544
F.27 JSON_Cap_CardReaderAntiSneakCfg .....	546
F.28 JSON_Cap_CardReaderCfg .....	547
F.29 JSON_Cap_ClearAntiSneak .....	552
F.30 JSON_Cap_ClearAntiSneakCfg .....	553
F.31 JSON_Cap_ClearPlansCfg .....	553
F.32 JSON_Cap_EventOptimizationCfg .....	554
F.33 JSON_Cap_FaceRecognizeMode .....	554
F.34 JSON_Cap_FingerPrintCfg .....	554
F.35 JSON_Cap_FingerPrintDelete .....	556
F.36 JSON_Cap_HealthCodeCfg .....	557
F.37 JSON_Cap_HealthCodeDisplayCfg .....	558
F.38 JSON_Cap_LogModeCfg .....	558
F.39 JSON_Cap OSDPModify .....	558
F.40 JSON_Cap OSDPStatus .....	559
F.41 JSON_Cap RegionCalibrationCfg .....	559
F.42 JSON_Cap RegionCoordinate .....	560
F.43 JSON_Cap RemoteCheck .....	560
F.44 JSON_Cap RemoteControlBuzzer .....	561
F.45 JSON_Cap TemperatureMeasurementCfg .....	561
F.46 JSON_Cap UserInfo .....	562
F.47 JSON_Cap UserInfoDetail .....	568
F.48 JSON_Cap UserRightHolidayGroupCfg .....	568

F.49 JSON_Cap_UserRightHolidayPlanCfg .....	569
F.50 JSON_Cap_UserRightPlanTemplate .....	569
F.51 JSON_Cap_UserRightWeekPlanCfg .....	570
F.52 JSON_CapturePreset .....	571
F.53 JSON_CapturePresetCap .....	571
F.54 JSON_CaptureProgress .....	571
F.55 JSON_CaptureRule .....	572
F.56 JSON_CaptureRuleCap .....	572
F.57 JSON_CardEncryption .....	573
F.58 JSON_CardInfo .....	573
F.59 JSON_CardInfo_Collection .....	574
F.60 JSON_CardInfoCap .....	575
F.61 JSON_CardInfoCount .....	575
F.62 JSON_CardInfoDelCond .....	575
F.63 JSON_CardInfoSearch .....	576
F.64 JSON_CardInfoSearchCond .....	577
F.65 JSON_CardOperationsCap .....	578
F.66 JSON_CardParam .....	586
F.67 JSON_CardProto .....	586
F.68 JSON_CardReaderAntiSneakCfg .....	587
F.69 JSON_CardReaderCfg .....	587
F.70 JSON_CardResetResponse .....	590
F.71 JSON_CardVerificationRule .....	590
F.72 JSON_CardVerificationRuleCap .....	591
F.73 JSON_CardVerificationRuleRes .....	591
F.74 JSON_ChannelControllerTypeCfg .....	592
F.75 JSON_ChannelControllerTypeCfgCap .....	592
F.76 JSON_ClearAntiSneak .....	592

F.77 JSON_ClearAntiSneakCfg .....	593
F.78 JSON_ClearAttendancePlan .....	593
F.79 JSON_ClearData .....	593
F.80 JSON_ClearDataRes .....	594
F.81 JSON_ClearPictureCfg .....	594
F.82 JSON_ClearPictureCfgCap .....	594
F.83 JSON_ClearPlansCfg .....	595
F.84 JSON_ControlBlock .....	596
F.85 JSON_CreateFPLibCond .....	596
F.86 JSON_CreateFPLibResult .....	597
F.87 JSON_CustomData .....	597
F.88 JSON_CustomDataRes .....	598
F.89 JSON_CustomDataResult .....	598
F.90 JSON_CustomDataSearchCond .....	598
F.91 JSON_DataBlock .....	599
F.92 JSON_DataBlockCtrl .....	599
F.93 JSON_DataOutputCfg .....	599
F.94 JSON_DataOutputProgress .....	600
F.95 JSON_DataTrans .....	600
F.96 JSON_DelFaceRecord .....	600
F.97 JSON_EditFplibInfo .....	601
F.98 JSON_EventNotificationAlert_AccessControllerEvent .....	601
F.99 JSON_EventNotificationAlert_Alarm/EventInfo .....	612
F.100 JSON_EventNotificationAlert_FaceTempScreeningEventMsg .....	613
F.101 JSON_EventNotificationAlert_QRCodeEventMsg .....	616
F.102 JSON_EventOptimizationCfg .....	619
F.103 JSON_EventStorageCfg .....	620
F.104 JSON_EventStorageCfgCap .....	620

F.105 JSON_FaceRecognizeMode .....	621
F.106 JSON_FaceRecordNumInAllFPLib .....	621
F.107 JSON_FaceRecordNumInOneFPLib .....	622
F.108 JSON_FaceTemperatureEvent .....	622
F.109 JSON_FaceTemperatureEventCap .....	624
F.110 JSON_FaceTemperatureEventCond .....	626
F.111 JSON_FingerPrintCfg .....	627
F.112 JSON_FingerPrintCond .....	628
F.113 JSON_FingerPrintDelete .....	628
F.114 JSON_FingerPrintDeleteProcess .....	629
F.115 JSON_FingerPrintInfo .....	629
F.116 JSON_FingerPrintModify .....	630
F.117 JSON_FingerPrintStatus .....	630
F.118 JSON_FPLibCap .....	631
F.119 JSON_FPLibListInfo .....	634
F.120 JSON_HealthCodeCfg .....	634
F.121 JSON_HealthCodeDisplayCfg .....	635
F.122 JSON_IDCardInfoEvent .....	635
F.123 JSON_IDCardInfoEventCap .....	638
F.124 JSON_IDCardInfoEventCond .....	643
F.125 JSON_IdentityInfo .....	644
F.126 JSON_IdentityInfoCap .....	645
F.127 JSON_IdentityInfoCond .....	647
F.128 JSON_IRCfg .....	647
F.129 JSON_IRCfgCap .....	648
F.130 JSON_LogModeCfg .....	648
F.131 JSON_MaskDetection .....	648
F.132 JSON_MaskDetectionCap .....	649

F.133 JSON_NFCCfg .....	649
F.134 JSON_NFCCfgCap .....	649
F.135 JSON_OfflineCaptureCap .....	650
F.136 JSON OSDPModify .....	657
F.137 JSON OSDPStatus .....	657
F.138 JSON_PersonInfoExtendName .....	658
F.139 JSON_PersonInfoExtendNameCap .....	658
F.140 JSON_PictureServerInformation .....	659
F.141 JSON_QRCodeEvent .....	660
F.142 JSON_QRCodeEventCap .....	661
F.143 JSON_QRCodeEventCond .....	664
F.144 JSON_RegionCalibrationCfg .....	665
F.145 JSON_RegionCoordinate .....	665
F.146 JSON_RemoteCheck .....	665
F.147 JSON_RemoteControlBuzzer .....	666
F.148 JSON_RemoteCtrlIlerModeCfg .....	666
F.149 JSON_RemoteCtrlIlerModeCfgCap .....	666
F.150 JSON_ResponseStatus .....	667
F.151 JSON_RFCardCfg .....	667
F.152 JSON_RFCardCfgCap .....	667
F.153 JSON_RuleInfo .....	668
F.154 JSON_SafetyHelmetDetection .....	669
F.155 JSON_SafetyHelmetDetectionCap .....	669
F.156 JSON_SearchFaceRecordCond .....	670
F.157 JSON_SearchFaceRecordResult .....	671
F.158 JSON_SearchTaskCond .....	672
F.159 JSON_SearchTaskResponse .....	673
F.160 JSON_SectionEncryption .....	676

F.161 JSON_SetFaceRecord .....	677
F.162 JSON_SingleFPLibInfo .....	678
F.163 JSON_TemperatureMeasurementCfg .....	679
F.164 JSON_UploadFailedDetails .....	680
F.165 JSON_UserInfo .....	680
F.166 JSON_UserInfoCount .....	683
F.167 JSON_UserInfoDelCond .....	683
F.168 JSON_UserInfoDetail .....	683
F.169 JSON_UserInfoDetailDeleteProcess .....	684
F.170 JSON_UserInfoSearch .....	684
F.171 JSON_UserInfoSearchCond .....	688
F.172 JSON_UserRightHolidayGroupCfg .....	688
F.173 JSON_UserRightHolidayPlanCfg .....	689
F.174 JSON_UserRightPlanTemplate .....	690
F.175 JSON_UserRightWeekPlanCfg .....	690
F.176 JSON_Verification .....	691
F.177 XML_AcsAbility .....	691
F.178 XML_CaptureFingerPrint .....	726
F.179 XML_CaptureFingerPrintCond .....	727
F.180 XML_Cap_AccessControl .....	727
F.181 XML_Cap_CaptureFingerPrint .....	737
F.182 XML_Cap_ChannelControllerCfg .....	737
F.183 XML_Cap_ClearCardRecord .....	738
F.184 XML_Cap_ClearSubmarineBack .....	739
F.185 XML_Cap_GetAcsEvent .....	739
F.186 XML_Cap_DeployInfo .....	741
F.187 XML_Cap_FaceCompareCond .....	741
F.188 XML_Cap_IdentityTerminal .....	742

F.189 XML_Cap_ReaderAcrossHost .....	743
F.190 XML_Cap_ServerDevice .....	744
F.191 XML_Cap_StartReaderInfo .....	744
F.192 XML_Cap_SubmarineBack .....	744
F.193 XML_Cap_SubmarineBackHostInfo .....	744
F.194 XML_Cap_SubmarineBackMode .....	745
F.195 XML_Cap_SubmarineBackReader .....	745
F.196 XML_ChannelControllerCfg .....	745
F.197 XML_ClearCardRecord .....	746
F.198 XML_ClearSubmarineBack .....	747
F.199 XML_DeployInfo .....	747
F.200 XML_DeviceCap .....	748
F.201 XML_Desc_AcsAbility .....	760
F.202 XML_EventNotificationAlert_AlarmEventInfo .....	760
F.203 XML_FaceCompareCond .....	761
F.204 XML_IdentityTerminal .....	762
F.205 XML_ReaderAcrossHost .....	763
F.206 XML_ResponseStatus .....	764
F.207 XML_ServerDevice .....	764
F.208 XML_StartReaderInfo .....	764
F.209 XML_SubmarineBack .....	765
F.210 XML_SubmarineBackHostInfo .....	765
F.211 XML_SubmarineBackMode .....	765
F.212 XML_SubmarineBackReader .....	766
F.213 XML_SubscribeEvent .....	766
<b>Appendix G. Response Codes of Text Protocol .....</b>	<b>768</b>
<b>Appendix H. Error Codes Categorized by Functional Modules .....</b>	<b>807</b>

# Chapter 1 Access Control

Access Control is the selective restriction of access to a place or other resources. The access control applications integrated by Device Network SDK (here is referred to as "HCNetSDK") in this manual take the person as the management and control unit, which indicates that all applications are integrated around the basic unit. That is, linking fingerprints, faces, and other attributes to a card will be replaced by linking fingerprints, cards, and other attributes to a person.

## 1.1 Introduction

This manual mainly introduces the integration flows and related APIs for access controller, fingerprint access control terminal, fingerprint time attendance terminal, and so on, to implement the following functions: schedule configuration, person/card/fingerprint information management, alarm/event configuration, door/elevator/buzzer control, anti-passing back, and so on.

## 1.2 Update History

### Summary of Changes in Version 6.1.7.15\_Aug., 2021

1. Extended the message about the face picture library capability [JSON\\_FPLibCap](#), the condition message of adding a face record to the face picture library [JSON\\_AddFaceRecordCond](#), the condition message about editing face records in the face picture library in a batch [JSON\\_BatchEditFaceRecord](#), and the condition message about setting the face record in the face picture library [JSON\\_SetFaceRecord](#) (related URIs: [/ISAPI/Intelligent/FDLib/capabilities?format=json](#) , [/ISAPI/Intelligent/FDLib/FaceDataRecord?format=json](#) , [/ISAPI/Intelligent/FDLib/FDModify?format=json](#) , and [/ISAPI/Intelligent/FDLib/FDSetUp?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
add a node **saveFacePic** (whether to save face pictures).
2. Extended the result message of searching for the face records in the a face picture library [JSON\\_SearchFaceRecordResult](#) (related URI: [/ISAPI/Intelligent/FDLib/FDSearch?format=json](#) ): added a sub node **saveFacePic** (whether to save face pictures) to the node **MatchList**.
3. Extended the configuration capability message [XML\\_Cap\\_IdentityTerminal](#) and the parameter message [XML\\_IdentityTerminal](#) of intelligent identity recognition terminal (related URIs: [/ISAPI/AccessControl/IdentityTerminal/capabilities](#) and [/ISAPI/AccessControl/IdentityTerminal](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a sub node **maskFaceMatchThreshold1** (1:1 face picture (face with mask and normal background picture) comparison threshold of ECO mode) to the node **ecoMode**.
4. Extended the configuration capability message [JSON\\_Cap\\_CardReaderCfg](#) and the parameter message [JSON\\_CardReaderCfg](#) of the card reader (related URIs: [/ISAPI/AccessControl/CardReaderCfg/capabilities?format=json](#) and [/ISAPI/AccessControl/CardReaderCfg/<ID>?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):

- added a node **maskFaceMatchThreshold1** (1:1 face picture (face with mask and normal background) comparison threshold).
5. Extended the configuration capability message **JSON\_Cap\_AcsCfg** and parameter message **JSON\_AcsCfg** of the access controller (related URIs: </ISAPI/AccessControl/AcsCfg/capabilities?format=json> and </ISAPI/AccessControl/AcsCfg?format=json> ; related API: **NET\_DVR\_STDXMLConfig**):  
added two nodes **desensitiseEmployeeNo** (whether to enable employee No. de-identification for local UI display) and **desensitiseName** (whether to enable name de-identification for local UI display).

## Summary of Changes in Version 6.1.7.5\_June., 2021

1. Extended the capability message of actively getting face temperature screening events **JSON\_FaceTemperatureEventCap** (related URI: </ISAPI/AccessControl/FaceTemperatureEvent/capabilities?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **isAbnormalTemperature** (whether the skin-surface temperature is abnormal) to the node **FaceTemperatureEventCond**.
2. Extended the condition message of actively getting face temperature screening events **JSON\_FaceTemperatureEventCond** (related URI: </ISAPI/AccessControl/FaceTemperatureEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node **isAbnormalTemperature** (whether the skin-surface temperature is abnormal).
3. Extended the capability message of searching for access control events **JSON\_Cap\_AcsEvent** (related URI: </ISAPI/AccessControl/AcsEvent/capabilities?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **isAbnormalTemperature** (whether the skin-surface temperature is abnormal) to the node **AcsEventCond**;  
added a sub node **HealthInfo** (health information) to the node **InfoList**.
4. Extended the condition message of searching for access control events **JSON\_AcsEventCond** (related URI: </ISAPI/AccessControl/AcsEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node **isAbnormalTemperature** (whether the skin-surface temperature is abnormal).
5. Extended the result message of searching for access control events **JSON\_AcsEvent** (related URI: </ISAPI/AccessControl/AcsEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **HealthInfo** (health information) to the node **InfoList**.
6. Extended the capability message of getting the ID card swiping events actively **JSON\_IDCardInfoEventCap** (related URI: </ISAPI/AccessControl/IDCardInfoEvent/capabilities?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **isAbnormalTemperature** (whether the skin-surface temperature is abnormal) to the node **IDCardInfoEventCond**;  
added a sub node **HealthInfo** (health information) to the node **InfoList**.
7. Extended the condition message of getting the ID card swiping events actively **JSON\_IDCardInfoEventCond** (related URI: </ISAPI/AccessControl/IDCardInfoEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node **isAbnormalTemperature** (whether the skin-surface temperature is abnormal).

8. Extended the result message of getting the ID card swiping events actively **JSON\_IDCardInfoEvent** (related URI: </ISAPI/AccessControl/IDCardInfoEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **HealthInfo** (health information) to the node **InfoList**.
9. Extended the configuration capability message **JSON\_Cap\_AcsCfg** and the parameter message **JSON\_AcsCfg** of the access controller (related URLs: </ISAPI/AccessControl/AcsCfg/capabilities?format=json> and </ISAPI/AccessControl/AcsCfg?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a verification channel type "ISAPIListen" (ISAPI listening channel) to the node **checkChannelType**;  
added a node **enableCaptureCertificate** (whether to enable capturing the ID picture).
10. Extended the functional capability message of access control **XML\_Cap\_AccessControl** (related URI: </ISAPI/AccessControl/capabilities> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added three nodes <isSupportAddCustomAudio> (whether it supports importing custom audio), <isSupportDeleteCustomAudio> (whether it supports deleting custom audio), and <isSupportSearchCustomAudio> (whether it supports searching for custom audio).
11. Extended the structure about access control event details **NET\_DVR\_ACSEVENT\_INFO** :  
added a member **byHealthCode** (health code status) by one byte.
12. Extended the structure about the alarm triggered by swiping ID card **NET\_DVR\_ID\_CARD\_INFO\_ALARM** :  
added a member **byHealthCode** (health code status) by one byte.
13. Extended the structure about access control event details **NET\_DVR\_ACSEVENT\_DETAIL** (related API: **NET\_DVR\_StartRemoteConfig** ):  
added a member **byHealthCode** (health code status) by one byte.

## Summary of Changes in Version 6.1.6.20\_Mar., 2021

1. Added functions of temperature measurement, refer to **Temperature Measurement** .
2. Added URIs of configuring health code parameters (related API: **NET\_DVR\_STDXMLConfig** ):  
Get configuration capability: GET </ISAPI/AccessControl/healthCodeCfg/capabilities?format=json> ;  
Get or set parameters: GET or PUT </ISAPI/AccessControl/healthCodeCfg?format=json> .
3. Added URIs of configuring health code display parameters (related API: **NET\_DVR\_STDXMLConfig** ):  
Get configuration capability: GET </ISAPI/AccessControl/showHealthCodeCfg/capabilities?format=json> ;  
Get or set parameters: GET or PUT </ISAPI/AccessControl/showHealthCodeCfg?format=json> .
4. Added two URIs of configuring black body parameters (related API: **NET\_DVR\_STDXMLConfig** ):  
Get configuration capability: GET </ISAPI/AccessControl/blackObject/capabilities?format=json> ;  
Get or set parameters: GET or PUT </ISAPI/AccessControl/blackObject?format=json> .
5. Extended the functional capability message of access control **XML\_Cap\_AccessControl** (related URI: </ISAPI/AccessControl/capabilities> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added six node: <isSupportTemperatureMeasureCfg> (whether it supports configuring temperature measurement parameters), <isSupportTemperatureMeasureAreaCfg> (whether it

- supports configuring parameters of the temperature measurement area), **<isSupportTemperatureMeasureAreaCalibrationCfg>** (whether it supports configuring calibration parameters of the temperature measurement area), **<isSupportBlackObjectCfg>** (whether it supports configuring black body parameters), **<isSupportHealthCodeCfg>** (whether it supports configuring health code parameters), and **<isSupportShowHealthCodeCfg>** (whether it supports configuring display parameters of the health code).
6. Extended the configuration capability message **JSON\_Cap\_AcsCfg** and the parameter message **JSON\_AcsCfg** of the access controller (related URLs: </ISAPI/AccessControl/AcsCfg/capabilities?format=json> and </ISAPI/AccessControl/AcsCfg?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added 7 nodes: **uploadVerificationPic** (whether to upload the authenticated picture), **saveVerificationPic** (whether to save the authenticated picture), **saveFacePic** (whether to save the registered face picture), **thermalUnit** (temperature unit), **highestThermalThresholdF** (the maximum value of the temperature threshold), **lowestThermalThresholdF** (the minimum value of the temperature threshold), and **thermalCompensation** (temperature compensation).
7. Extended the event information message of face temperature screening  
**JSON\_EventNotificationAlert\_FaceTempScreeningEventMsg** :  
added a sub node **helmet** (whether the person wears a hard hat) to the node **FaceTemperatureMeasurementEvent**.
8. Extended the event message of scanning QR code  
**JSON\_EventNotificationAlert\_QRCodeEventMsg** :  
added a sub node **helmet** (whether the person wears a hard hat) to the node **QRCodeEvent**.
9. Extended the configuration capability message **XML\_Cap\_IdentityTerminal** and the parameter message **XML\_IdentityTerminal** of the intelligent identity recognition terminal (related URLs: </ISAPI/AccessControl/IdentityTerminal/capabilities> and </ISAPI/AccessControl/IdentityTerminal> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node <showMode> (display mode).
10. Extended the capability message **JSON\_FaceTemperatureEventCap** and the result message **JSON\_FaceTemperatureEvent** of actively getting face temperature screening events (related URLs: </ISAPI/AccessControl/FaceTemperatureEvent/capabilities?format=json> and </ISAPI/AccessControl/FaceTemperatureEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **helmet** (whether the person wears a hard hat) to the node **InfoList**.
11. Extended the capability message **JSON\_QRCodeEventCap** and the result message **JSON\_QRCodeEvent** of actively getting QR code scanning events (related URLs: </ISAPI/AccessControl/QRCodeEvent/capabilities?format=json> and </ISAPI/AccessControl/QRCodeEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **helmet** (whether the person wears a hard hat) to the node **InfoList**.
12. Extended the capability message **JSON\_IDCardInfoEventCap** and the result message **JSON\_IDCardInfoEvent** of actively getting ID card swiping events (related URLs: </ISAPI/AccessControl/IDCardInfoEvent/capabilities?format=json> and </ISAPI/AccessControl/IDCardInfoEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **helmet** (whether the person wears a hard hat) to the node **InfoList**.

## Summary of Changes in Version 6.1.6.20\_Jan., 2021

1. Extended face picture library capability message [JSON\\_FPLibCap](#) (related URI: [/ISAPI/Intelligent/FDLib/capabilities?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a node **featurePointTypeList** (feature point types of face pictures supported by the device).
2. Extended the condition message of setting the face record [JSON\\_SetFaceRecord](#) (related URI: [/ISAPI/Intelligent/FDLib/FDSetUp?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a node **PicFeaturePoints** (feature points to be applied).
3. Extended the condition message of editing face records in the face picture library in a batch [JSON\\_BatchEditFaceRecord](#) (related URI: [/ISAPI/Intelligent/FDLib/FDModify?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a node **PicFeaturePoints** (feature points to be applied).
4. Extended the condition message of adding a face record to the face picture library [JSON\\_AddFaceRecordCond](#) (related URI: [/ISAPI/Intelligent/FDLib/FaceDataRecord?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a node **PicFeaturePoints** (feature points to be applied).
5. Extended the result message of searching for person information [JSON\\_UserInfoSearch](#) (related URI: [/ISAPI/AccessControl/UserInfo/Search?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a sub node **id** (ID of the additional person information) to the node **PersonInfoExtends** of **Userinfo**;  
deleted a sub node **name** from the node **PersonInfoExtends** of **Userinfo**.
6. Extended the person information message [JSON\\_UserInfo](#) (related URIs: [/ISAPI/AccessControl/UserInfo/Modify?format=json](#) , [/ISAPI/AccessControl/UserInfo/Record?format=json](#) , and [/ISAPI/AccessControl/UserInfo/SetUp?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a sub node **id** (ID of the additional person information) to the node **PersonInfoExtends**;  
deleted a sub node **name** from the node **PersonInfoExtends**.
7. Extended the person management capability message [JSON\\_Cap\\_UserInfo](#) (related URI: [/ISAPI/AccessControl/UserInfo/capabilities?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a sub node **id** (ID of the additional person information) to the node **PersonInfoExtends**;  
deleted a sub node **name** from the node **PersonInfoExtends**.
8. Added two URIs of configuring the name of the additional person information (related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
Get configuration capability: GET [/ISAPI/AccessControl/personInfoExtendName/capabilities?format=json](#) ;  
Get or set parameters: GET or PUT [/ISAPI/AccessControl/personInfoExtendName?format=json](#) .
9. Extended the structure about alarms triggered by swiping ID card  
[NET\\_DVR\\_ID\\_CARD\\_INFO\\_ALARM](#) :  
added a member **byHelmet** (whether the person is wearing a hard hat) by one byte.
10. Extended log types in [HCNetSDK Log Types](#) :

added 15 operation log types: 0x3002—"MINOR\_LOCAL\_PARA\_FACTORY\_DEFAULT" (Restore to default settings locally), 0x3003—"MINOR\_REMOTE\_PARA\_FACTORY\_DEFAULT" (Restore to default settings remotely), 0x3004—"MIMOR\_REMOTE\_DELETE\_ALL\_VERIFYORCAP\_PICS" (Delete all authenticated or captured face pictures remotely), 0x3005—"MIMOR\_LOCAL\_DELETE\_ALL\_VERIFYORCAP\_PICS" (Delete all authenticated or captured face pictures locally), 0x3006—"MIMOR\_REMOTE\_DELETE\_EVENTS\_AT\_SPECTIME" (Delete events by specified time remotely), 0x3007—"MIMOR\_LOCAL\_DELETE\_EVENTS\_AT\_SPECTIME" (Delete events by specified time locally), 0x3008—"MIMOR\_REMOTE\_OPEN\_SUMMER\_TIME" (Enable DST remotely), 0x3009—"MIMOR\_LOCAL\_OPEN\_SUMMER\_TIME" (Enable DST locally), 0x3010—"MIMOR\_REMOTE\_CLOSE\_SUMMER\_TIME" (Disable DST remotely), 0x3011—"MIMOR\_LOCAL\_CLOSE\_SUMMER\_TIME" (Disable DST locally), 0x3012—"MIMOR\_REMOTE\_EZVIZ\_UNBIND" (Unbind from EZVIZ cloud remotely), 0x3013—"MIMOR\_LOCAL\_EZVIZ\_UNBIND" (Unbind from EZVIZ cloud locally), 0x3014—"MIMOR\_ENTER\_LOCALUI\_BACKGROUND" (Enter UI background), 0x3015—"MIMOR\_REMOTE\_DELETE\_FACEBASEMAP" (Delete registered face pictures remotely), and 0x3016—"MIMOR\_LOCAL\_DELETE\_FACEBASEMAP" (Delete registered face pictures locally);  
added four additional information log types: 0x432—"MINOR\_ADD\_USER\_INFO" (Added person information (access control permission)), 0x433—"MINOR MODIFY\_USER\_INFO" (Edit person information (access control permission)), 0x434—"MINOR\_CLR\_USER\_INFO" (Delete person information by employee No. (access control permission)), and 0x435—"MINOR\_CLR\_CARD\_BY\_CARD\_OR\_EMPLOYEE" (Delete cards by card No. or employee No.).

### Summary of Changes in Version 6.1.5.20\_Oct., 2020

1. Added two query parameters **security** (the version No. of encryption scheme) and **iv** (the initialization vector) to the request URI [/ISAPI/AccessControl/CapturePresetParam?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).
2. Added two query parameters **security** (the version No. of encryption scheme) and **iv** (the initialization vector) to the request URI [/ISAPI/AccessControl/CaptureCardInfo?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).
3. Extended the capability message of collecting card information [JSON\\_CardInfoCap](#) and card information message [JSON\\_CardInfo\\_Collection](#) (related URIs: [/ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json](#) and [/ISAPI/AccessControl/CaptureCardInfo?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added two card types "FelicaCard" (Felica card) and "DesfireCard" (DESFire card) to the node **cardType**.
4. Added two query parameters **security** (the version No. of encryption scheme) and **iv** (the initialization vector) to the request URI [/ISAPI/AccessControl/CaptureIDInfo?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).

5. Added a URI of getting details of failing to upload the user list of offline collection (related API: [NET\\_DVR\\_STDXMLConfig](#)): GET [/ISAPI/AccessControl/OfflineCapture/uploadFailedDetails?format=json](#).
6. Extended the result message of searching for the collected data [JSON\\_SearchTaskResponse](#) (related URI: [/ISAPI/AccessControl/OfflineCapture/DataCollections/searchTask?format=json](#); related API: [NET\\_DVR\\_StartRemoteConfig](#)):  
added two sub nodes **cardNo** (card No.) and **cardType** (card type) to the node **CardNoList** of **DataCollections**;  
added two sub nodes **IdentityInfo** (identity information) and **CardIssueStatus** (issuing status list of cards containing face pictures and fingerprints) to the node **DataCollections**.
7. Extended parameter message of offline collection rules [JSON\\_RuleInfo](#) (related URI: [/ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added two nodes **enableLocalIssueCard** (whether to enable issuing smart cards locally) and **isLocalStorage** (whether to store face picture and fingerprint information in the device locally).
8. Extended parameter message of offline collection progress [JSON\\_CaptureProgress](#) (related URI: [/ISAPI/AccessControl/OfflineCapture/progress?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added two nodes **reqIssueNum** (number of persons to be issued with smart cards) and **IssuedNum** (number of persons that have been issued with smart cards).
9. Added two query parameters **security** (the version No. of encryption scheme) and **iv** (the initialization vector) to the request URI [/ISAPI/AccessControl/OfflineCapture/dataOutput?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#)).
10. Extended parameter message for exporting offline collected data [JSON\\_DataOutputCfg](#) (related URI: [/ISAPI/AccessControl/OfflineCapture/dataOutput?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added a node **type** (exporting type).
11. Extended the offline collection capability message [JSON\\_OfflineCaptureCap](#) (related URI: [/ISAPI/AccessControl/OfflineCapture/capabilities?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added three sub nodes **maxSize** (size of the card No. list), **cardNo** (card No.), and **cardType** (card type) to the node **CardNoList** of **DataCollections** of **SearchTask**;  
added two sub nodes **IdentityInfo** (identity information) and **CardIssueStatus** (issuing status list of cards containing face pictures and fingerprints) to the node **DataCollections** of **SearchTask**;  
added two nodes **enableLocalIssueCard** (whether to enable issuing smart cards locally) and **isLocalStorage** (whether to store face picture and fingerprint information in the device locally) to the node **RuleInfo**;  
added two nodes **reqIssueNum** (number of persons to be issued with smart cards) and **IssuedNum** (number of persons that have been issued with smart cards) to the node **CaptureProgress**.
12. Added two query parameters **security** (the version No. of encryption scheme) and **iv** (the initialization vector) to the request URI [/ISAPI/AccessControl/CardOperations/sectionEncryption?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#)).

13. Added two query parameters **security** (the version No. of encryption scheme) and **iv** (the initialization vector) to the request URI [/ISAPI/AccessControl/CardOperations/verification?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).
14. Added two query parameters **security** (the version No. of encryption scheme) and **iv** (the initialization vector) to the request URI [/ISAPI/AccessControl/CardOperations/controlBlock?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).
15. Added a URI of deleting data from the card: PUT [/ISAPI/AccessControl/CardOperations/clearData?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).
16. Added a URI of setting custom card information: PUT [/ISAPI/AccessControl/CardOperations/customData?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).
17. Added a URI of searching for custom card information: POST [/ISAPI/AccessControl/CardOperations/customData/searchTask?format=json](#) (related API: [NET\\_DVR\\_STDXMLConfig](#) ).
18. Extended card operation capability message [JSON\\_CardOperationsCap](#) (related URI: [/ISAPI/AccessControl/CardOperations/capabilities?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added seven nodes: **Issue** (capability of sending a request for card issuing and getting the current card issuing status and real-time card issuing results), **localIssueCfg** (capability of configuring rule parameters for issuing smart cards), **ClearData** (capability of deleting data from the card), **CustomData** (capability of setting custom card information), **CustomDataSearchCond** (condition configuration capability of searching for custom card information), **CustomDataResult** (result capability of searching for custom card information), and **CardIssueStatus** (capability of getting the smart card issuing status).
19. Added 10 additional information logs to [HCNetSDK Log Types](#) :  
0x423-"MINOR\_USB\_LOGIN" (Log in via USB), 0x424-"MINOR\_USB\_LOGOUT" (Log out via USB), 0x425-"MINOR\_ISAPI\_HTTP\_LOGIN" (Log in via ISAPI (HTTP)),  
0x426-"MINOR\_ISAPI\_HTTP\_LOGOUT" (Log out via ISAPI (HTTP)),  
0x427-"MINOR\_ISAPI\_HTTPS\_LOGIN" (Log in via ISAPI (HTTPS)),  
0x428-"MINOR\_ISAPI\_HTTPS\_LOGOUT" (Log out via ISAPI (HTTPS)),  
0x429-"MINOR\_ISUP\_ONLINE" (ISUP online), 0x42a-"MINOR\_ISUP\_OFFLINE" (ISUP offline),  
0x42b-"MINOR\_FP\_ISSUE\_REC" (Issuing record of card containing fingerprint information), and  
0x42c-"MINOR\_FACE\_ISSUE\_REC" (Issuing record of card containing face picture information).

### Summary of Changes in Version 6.1.5.15\_Aug., 2020

1. Extended the functional capability message of access control [XML\\_Cap\\_AccessControl](#) (related URI: [/ISAPI/AccessControl/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added 9 nodes, i.e., **<isSupportSafetyHelmetDetection>** (whether it supports configuring hard hat detection), **<isSupportKeyCfgAttendance>** (whether it supports configuring parameters of attendance check by pressing the key), **<isSupportIDBlackListTemplate>** (whether it supports downloading the ID card blocklist template), **<isSupportAttendanceWeekPlan>** (whether it supports configuring parameters of the week attendance schedule),  
**<isSupportClearAttendancePlan>** (whether it supports clearing the week attendance schedule),  
**<isSupportAttendanceMode>** (whether it supports configuring the attendance mode),

- <**isSupportAttendancePlanTemplate**> (whether it supports configuring the attendance schedule template), <**isSupportAttendancePlanTemplateList**> (whether it supports getting the list of attendance schedule templates), and <**isSupportCardVerificationRule**> (whether it supports configuring card No. authentication mode).
2. Extended capability message **JSON\_Cap\_AcsEvent** and result parameter message **JSON\_AcsEvent** of searching for access control events (related URIs: [/ISAPI/AccessControl/AcsEvent/capabilities?format=json](#) and [/ISAPI/AccessControl/AcsEvent?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added three sub nodes to the node **InfoList**, i.e., **label** (custom attendance name), **mask** (whether the person is wearing mask), and **helmet** (whether the person is wearing hard hat).
3. Extended the configuration capability message **JSON\_Cap\_CardReaderCfg** and the parameter message **JSON\_CardReaderCfg** of the card reader (related URIs: [/ISAPI/AccessControl/CardReaderCfg/capabilities?format=json](#) and [/ISAPI/AccessControl/CardReaderCfg/<ID>?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added three nodes: **enableReverseCardNo** (whether to enable reversing the card No.), **independSwipeIntervals** (time interval of person authentication), and **maskFaceMatchThresholdN** (1:N face picture (face with mask and normal background) comparison threshold).
4. Extended message about access control event information  
**JSON\_EventNotificationAlert\_AccessControllerEvent** :  
added two sub nodes **label** (custom attendance name) and **helmet** (whether the person is wearing hard hat) to the node **AccessControllerEvent**.
5. Extended configuration capability message **XML\_Cap\_IdentityTerminal** and parameter message **XML\_IdentityTerminal** of intelligent identity recognition terminal (related URIs: [/ISAPI/AccessControl/IdentityTerminal/capabilities](#) and [/ISAPI/AccessControl/IdentityTerminal](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a sub node <**maskFaceMatchThresholdN**> (1:N face picture (face with mask and normal background picture) comparison threshold of ECO mode) to the node <**ecoMode**>.
6. Extended configuration capability message **JSON\_RFCardCfgCap** and parameter message **JSON\_RFCardCfg** of enabling RF (Radio Frequency) card recognition (related URIs: [/ISAPI/AccessControl/Configuration/RFCardCfg/capabilities?format=json](#) and [/ISAPI/AccessControl/Configuration/RFCardCfg?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
add two card types "DesfireCard" (DESFire card) and "FelicaCard" (FeliCa card) to the node **cardType**.
7. Added the function of configuring attendance status and schedule, refer to [Configure Attendance Status and Schedule](#) .
8. Added URIs of configuring card No. authentication mode (related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
Get the configuration capability: GET [/ISAPI/AccessControl/CardVerificationRule/capabilities?format=json](#) ;  
Get or set parameters: GET or PUT [/ISAPI/AccessControl/CardVerificationRule?format=json](#) ;

- Get the switching progress and configuration result: GET [/ISAPI/AccessControl/CardVerificationRule/progress?format=json](#).
9. Extended structure about extended access control event information (V20)  
**NET\_DVRACS\_EVENT\_INFO\_EXTEND\_V20**:  
added a member **byAttendanceLabel** (custom attendance name) by 64 bytes.

### Summary of Changes in Version 6.1.5.10\_July, 2020

1. Extended configuration capability message **XML\_Cap\_ChannelControllerCfg** and parameter message **XML\_ChannelControllerCfg** of the lane controller (related URIs: [/ISAPI/AccessControl/ChannelControllerCfg/capabilities](#) and [/ISAPI/AccessControl/ChannelControllerCfg](#); related API: **NET\_DVR\_STDXMLConfig**):  
added a node <runMode> (running mode).
2. Added two URIs of configuring parameters of the keyfob control mode (related API: **NET\_DVR\_STDXMLConfig**):  
Get configuration capability: GET [/ISAPI/AccessControl/remoteControllerModeCfg/capabilities?format=json](#);  
Get or set parameters: GET or PUT [/ISAPI/AccessControl/remoteControllerModeCfg?format=json](#).
3. Extended functional capability message of access control **XML\_Cap\_AccessControl** (related URI: [/ISAPI/AccessControl/capabilities](#); related API: **NET\_DVR\_STDXMLConfig**):  
added a node <isSupportRemoteControllerModeCfg> (whether it supports configuring parameters of the keyfob control mode).

### Summary of Changes in Version 6.1.4.16\_Apr., 2020

1. Extended access control alarm/event information structure **NET\_DVRACS\_ALARM\_INFO**:  
added two members: **byAcsEventInfoExtendV20** (whether the member **pAcsEventInfoExtendV20** is valid) and **pAcsEventInfoExtendV20** (pointer of the structure **NET\_DVRACS\_EVENT\_INFO\_EXTEND\_V20**) by five bytes.
2. Extended structure about access control event details **NET\_DVRACS\_EVENT\_INFO**:  
added a member **byMask** (whether the person is wearing mask) by one byte.
3. Extended structure about the alarm triggered by swiping ID card  
**NET\_DVR\_ID\_CARD\_INFO\_ALARM**:  
added four members: **byMask** (whether the person is wearing mask), **byCurrentEvent** (whether it is a real-time event), **byIDCardInfoExtend** (whether the member **pIDCardInfoExtend** is valid), and **pIDCardInfoExtend** (pointer of the structure **NET\_DVR\_ID\_CARD\_INFO\_EXTEND**) by 7 bytes.
4. Added a message about event information of scanning QR code (command: 0x6009-"COMM\_ISAPI\_ALARM") **JSON\_EventNotificationAlert\_QRCodeEventMsg**.
5. Added a message about face temperature screening event information (command: 0x6009-"COMM\_ISAPI\_ALARM") **JSON\_EventNotificationAlert\_FaceTempScreeningEventMsg**.
6. Added two URIs of verifying the access control event remotely (related API: **NET\_DVR\_STDXMLConfig**):  
Get capability: GET [/ISAPI/AccessControl/remoteCheck/capabilities?format=json](#);

- Verify the access control event remotely: PUT [/ISAPI/AccessControl/remoteCheck?format=json](#).
7. Extended configuration capability message of the access controller [JSON\\_AcsCfg](#) (related URI: [/ISAPI/AccessControl/AcsCfg/capabilities?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added 11 nodes: **thermalEnabled** (whether to enable temperature measurement), **thermalMode** (whether to enable temperature measurement only mode), **thermalPictureEnabled** (whether to enable uploading visible light pictures in temperature measurement only mode), **isSupportThermalIp** (whether it supports configuring IP address of the thermography device), **highestThermalThreshold** (upper limit of the temperature threshold), **lowestThermalThreshold** (lower limit of the temperature threshold), **thermalDoorEnabled** (whether to open the door when the temperature is above the upper limit or below the lower limit of the threshold), **QRCodeEnabled** (whether to enable QR code function), **remoteCheckDoorEnabled** (whether to enable controlling the door by remote verification), **checkChannelType** (verification channel type), and **isSupportChannelIp** (whether it supports configuring IP address of the verification channel).
8. Extended parameter message of the access controller [JSON\\_AcsCfg](#) (related URI: [/ISAPI/AccessControl/AcsCfg?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added 11 nodes: **thermalEnabled** (whether to enable temperature measurement), **thermalMode** (whether to enable temperature measurement only mode), **thermalPictureEnabled** (whether to enable uploading visible light pictures in temperature measurement only mode), **thermalIp** (IP address of the thermography device), **highestThermalThreshold** (upper limit of the temperature threshold), **lowestThermalThreshold** (lower limit of the temperature threshold), **thermalDoorEnabled** (whether to open the door when the temperature is above the upper limit or below the lower limit of the threshold), **QRCodeEnabled** (whether to enable QR code function), **remoteCheckDoorEnabled** (whether to enable controlling the door by remote verification), **checkChannelType** (verification channel type), and **channelIp** (IP address of the verification channel).
9. Added two URIs of configuring mask detection parameters (related API: [NET\\_DVR\\_STDXMLConfig](#)):  
Get configuration capability: GET [/ISAPI/AccessControl/maskDetection/capabilities?format=json](#);  
Get or set parameters: GET or PUT [/ISAPI/AccessControl/maskDetection?format=json](#).
10. Extended structure about access control event details [NET\\_DVR\\_ACSEVENT\\_DETAIL](#) (related API: [NET\\_DVR\\_StartRemoteConfig](#)):  
added five members: **byMask** (whether the person is wearing mask or not), **byThermometryUnit** (temperature unit), **byIsAbnormalTemperature** (whether the face temperature is abnormal), **fCurrTemperature** (face temperature), and **strRegionCoordinates** (face temperature's coordinates) by 15 bytes.
11. Extended structure about access control event parameters [NET\\_DVR\\_ACSEVENT\\_CFG](#) (related API: [NET\\_DVR\\_StartRemoteConfig](#)):  
added six members: **dwQRCodeInfoLen** (length of the QR code information), **dwVisibleLightDataLen** (length of the visible light picture captured by the thermal camera),

- dwThermalDataLen** (length of the thermal picture), **pQRCodeInfo** (pointer of the QR code information), **pVisibleLightData** (pointer of the visible light picture captured by the thermal camera), and **pThermalData** (pointer of the thermal picture) by 24 bytes.
12. Extended access control capability message [XML\\_Cap\\_AccessControl](#) (related URI: [/ISAPI/AccessControl/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added two nodes: <isSupportRemoteCheck> (whether it supports verifying access control events remotely) and <isSupportMaskDetection> (whether it supports mask detection).
13. Extended device capability message [XML\\_DeviceCap](#) (related URI: [/ISAPI/System/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added two nodes: <isSupportFaceTemperatureMeasurementEvent> (whether it supports uploading face temperature screening events) and <isSupportQRCodeEvent> (whether it supports uploading QR code events).

## Summary of Changes in Version 6.1.3.40\_Feb., 2020

1. Extended person management capability message [JSON\\_Cap\\_UserInfo](#) (related URI: [/ISAPI/AccessControl/UserInfo/capabilities?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a node **purePwdVerifyEnable** (whether the device supports opening the door only by password).
2. Extended condition message of searching for access control events [JSON\\_AcsEventCond](#) (related URI: [/ISAPI/AccessControl/AcsEvent?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a node **timeReverseOrder** (whether to return events in descending order of time).
3. Extended capability message of searching for access control events [JSON\\_Cap\\_AcsEvent](#) (related URI: [/ISAPI/AccessControl/AcsEvent/capabilities?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a sub node **timeReverseOrder** (whether to return events in descending order of time) to the node **AcsEventCond**.
4. Extended configuration capability message [JSON\\_Cap\\_CardReaderCfg](#) and parameter message [JSON\\_CardReaderCfg](#) of card reader (related URIs: [/ISAPI/AccessControl/CardReaderCfg/capabilities?format=json](#) and [/ISAPI/AccessControl/CardReaderCfg/<ID>?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added two nodes: **FPAlgorithmVersion** (fingerprint algorithm library version) and **cardReaderVersion** (card reader version).
5. Extended structure about extended access control event information  
[NET\\_DVR\\_ACS\\_EVENT\\_INFO\\_EXTEND](#) :  
added a member **byPurePwdVerifyEnable** (whether the device supports opening the door only by password) by one byte.
6. Extended access control capability [XML\\_AcsAbility](#) (related API: [NET\\_DVR\\_GetDeviceAbility](#) ; capability type: 0x801-"ACS\_ABILITY"):  
added a sub node <**purePwdVerifyEnable**> (whether the device supports opening the door only by password) to the node <**WeekPlan**> of <**CardReaderVerifyTypePlan**> and the node <**EventLinkage**>, respectively;

added an event type "PasswordVerifyPass" (password authenticated) to the sub node <EventEntry> (index: 3) of the node <EventLinkage> (event card linkage).

### 7. Extended access control event types in **Access Control Event Types** :

added six exception event types to MAJOR\_EXCEPTION: "MINOR\_AUXILIARY\_BOARD\_OFFLINE" (Auxiliary Board Disconnected), "MINOR\_AUXILIARY\_BOARD\_RESUME" (Auxiliary Board Connected), "MINOR\_IDCARD\_SECURITY\_MOUDLE\_EXCEPTION" (Secure ID Card Unit Exception), "MINOR\_IDCARD\_SECURITY\_MOUDLE\_RESUME" (Secure ID Card Unit Restored), "MINOR\_FP\_PERIPHERAL\_EXCEPTION" (Fingerprint Collection Peripheral Exception), and "MINOR\_FP\_PERIPHERAL\_RESUME" (Fingerprint Collection Peripheral Restored);  
added three operation event types to MAJOR\_OPERATION: "MINOR\_OFFLINE\_DATA\_OUTPUT" (Export Offline Collected Data), "MINOR\_CREATE\_SSH\_LINK" (Establish SSH Connection), and "MINOR\_CLOSE\_SSH\_LINK" (Disconnect SSH Connection);  
added one event type to MAJOR\_EVENT: "MINOR\_PASSWD\_VERIFY\_PASS" (Password Authenticated).

### 8. Extended log types in **HCNetSDK Log Types** :

added 20 minor types to additional information logs (MAJOR\_INFORMATION-0x4):  
0x40e-"MINOR\_CLR\_USER" (clear all users), 0x40f-"MINOR\_CLR\_CARD" (clear all cards),  
0x410-"MINOR\_CLR\_FINGER\_BY\_READER" (clear all fingerprints by fingerprint and card reader),  
0x411-"MINOR\_CLR\_FINGER\_BY\_CARD" (clear all fingerprints by card No.),  
0x412-"MINOR\_CLR\_FINGER\_BY\_EMPLOYEE\_ON" (clear all fingerprints by employee ID),  
0x413-"MINOR\_DEL\_FINGER" (delete a fingerprint), 0x414-"MINOR\_CLR\_WEEK\_PLAN" (clear week schedules of access permission control), 0x415-"MINOR\_SET\_WEEK\_PLAN" (set the week schedule of access permission control), 0x416-"MINOR\_SET\_HOLIDAY\_PLAN" (set the holiday schedule of access permission control), 0x417-"MINOR\_CLR\_HOLIDAY\_PLAN" (clear holiday schedules of access permission control), 0x418-"MINOR\_SET\_HOLIDAY\_GROUP" (set the holiday group of access permission control schedule), 0x419-"MINOR\_CLR\_HOLIDAY\_GROUP" (clear holiday groups of access permission control schedule), 0x41a-"MINOR\_CLR\_TEMPLATE\_PLAN" (clear access permission control schedules), 0x41b-"MINOR\_SET\_TEMPLATE\_PLAN" (set the access permission control schedule), 0x41c-"MINOR\_ADD\_CARD" (add a card),  
0x41d-"MINOR\_MOD\_CARD" (edit a card), 0x41e-"MINOR\_ADD\_FINGER\_BY\_CARD" (add a fingerprint by card No.), 0x41f-"MINOR\_ADD\_FINGER\_BY\_EMPLOYEE\_NO" (add a fingerprint by employee ID), 0x420-"MINOR\_MOD\_FINGER\_BY\_CARD" (edit a fingerprint by card No.), and  
0x421-"MINOR\_MOD\_FINGER\_BY\_EMPLOYEE\_NO" (edit a fingerprint by employee ID).

## Summary of Changes in Version 6.1.3.10\_Jan., 2020

1. Extended person information message **JSON UserInfo** (related URIs: [/ISAPI/AccessControl/UserInfo/Record?format=json](#) , [/ISAPI/AccessControl/UserInfo/Modify?format=json](#) , and [/ISAPI/AccessControl/UserInfo/SetUp?format=json](#) ; related API: [\*\*NET\\_DVR\\_STDXMLConfig\*\*](#) ): added two nodes: **gender** (gender of the person in the face picture) and **PersonInfoExtends** (person extension information).
2. Extended result message of searching for person information **JSON UserInfoSearch** (related URI: [/ISAPI/AccessControl/UserInfo/Search?format=json](#) ; related API: [\*\*NET\\_DVR\\_STDXMLConfig\*\*](#) ):

- added two sub nodes: **gender** (gender of the person in the face picture) and **PersonInfoExtends** (person extension information) to the node **UserInfo**.
3. Extended person management capability message [JSON\\_Cap\\_UserInfo](#) (related URI: [/ISAPI/AccessControl/UserInfo/capabilities?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ): added a sub node **fuzzySearch** (keywords for fuzzy search) to the node **UserInfoSearchCond**; added two nodes: **gender** (gender of the person in the face picture) and **PersonInfoExtends** (person extension information).
  4. Extended condition configuration capability [XML\\_Cap\\_FaceCompareCond](#) and condition parameter message [XML\\_FaceCompareCond](#) of face picture comparison (related URLs: [/ISAPI/AccessControl/FaceCompareCond/capabilities](#) and [/ISAPI/AccessControl/FaceCompareCond](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ): added a node <**maxDistance**> (maximum recognition distance).
  5. Extended functional capability message of access control [XML\\_Cap\\_AccessControl](#) (related URI: [/ISAPI/AccessControl/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ): added six nodes: <**isSupportTTSText**> (whether it supports configuring the text of the audio prompt), <**isSupportIDBlackListCfg**> (whether it supports applying ID card blocklist), <**isSupportUserDataImport**> (whether it supports importing person permission data), <**isSupportUserDataExport**> (whether it supports exporting person permission data), <**isSupportMaintenanceDataExport**> (whether it supports exporting maintenance data), and <**isSupportLockTypeCfg**> (whether it supports configuring door lock status when the device is powered off).

## Summary of Changes in Version 6.1.3.X\_Sep., 2019

1. Extended access control capability message [XML\\_Cap\\_AccessControl](#) (related URL: [/ISAPI/AccessControl/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ): added five nodes: <**isSupportCaptureIDInfo**> (whether it supports collecting ID card information), <**isSupportCaptureRule**> (whether it supports configuring online collection rules), <**isSupportCapturePresetParam**> (whether it supports configuring preset parameters of online collection), <**isSupportOfflineCapture**> (whether it supports offline collection), and <**isSupportCardOperations**> (whether it supports card operation).
2. Added the function of online collecting data, refer to [Online Collect Data](#) .
3. Extended capability message [JSON\\_CardInfoCap](#) and parameter message [JSON\\_CardInfo\\_Collection](#) of collecting card information (related URLs: [/ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json](#) and [/ISAPI/AccessControl/CaptureCardInfo?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ): added a node **cardType** (card type).
4. Extended collected face data structure [NET\\_DVR\\_CAPTURE\\_FACE\\_CFG](#) (related API: [NET\\_DVR\\_StartRemoteConfig](#) ): added three members: **byFacePicQuality** (face quality in the face picture), **byInfraredFacePicQuality** (face quality in the infrared face picture), and **strcFeature** (feature information in the matted face picture) by 58 bytes.
5. Extended structure about picture data in JSON format [NET\\_DVR\\_JSON\\_DATA\\_CFG](#) (related API: [NET\\_DVR\\_SendRemoteConfig](#) ):

- added two members: **dwInfraredFacePicSize** (data size of infrared face picture) and **lpInfraredFacePicBuffer** (buffer of infrared face picture data) by eight bytes.
6. Added the function of offline collecting data, refer to [\*\*\*Offline Collect Data\*\*\*](#).
  7. Added three error codes to [\*\*\*Device Network SDK Errors\*\*\*](#): 1927-"NET\_ERR\_CAPTURE\_TIMEOUT" (collection timed out), 1928-"NET\_ERR\_LOW\_SCORE" (low quality of collected data), and 1929-"NET\_ERR\_OFFLINE\_CAPTURING" (the device is collecting data offline and cannot respond).
  8. Added two sub status codes: 0x30006000-"captureTimeout" (data collection timed out) and 0x30006001-"lowScore" (low quality of collected data) to status code 3 (Device Error) in [\*\*\*Response Codes of Text Protocol\*\*\*](#).
  9. Added functions of operating cards, refer to [\*\*\*Card Operation\*\*\*](#) for details.
  10. Added functions of configuring active infrared intrusion parameters (related API: [\*\*\*NET\\_DVR\\_STDXMLConfig\*\*\*](#)):  
Get configuration capability: GET [\*\*\*/ISAPI/AccessControl/Configuration/IRCfg/capabilities?format=json\*\*\*](#)  
Get or set parameters: GET or PUT [\*\*\*/ISAPI/AccessControl/Configuration/IRCfg?format=json\*\*\*](#)
  11. Added multiple log types, refer to [\*\*\*HCNetSDK Log Types\*\*\*](#) for details:  
added six minor log types to the "MAJOR\_EXCEPTION" log type:  
MINOR\_AUXILIARY\_BOARD\_OFFLINE (0x43c), MINOR\_AUXILIARY\_BOARD\_RESUME (0x43d),  
MINOR\_IDCARD\_SECURITY\_MOUDLE\_EXCEPTION (0x43e),  
MINOR\_IDCARD\_SECURITY\_MOUDLE\_RESUME (0x43f), MINOR\_FP\_PERIPHERAL\_EXCEPTION (0x440), and MINOR\_FP\_PERIPHERAL\_RESUME (0x441);  
added three minor log types to the "MAJOR\_OPERATION" log type:  
MINOR\_OFFLINE\_DATA\_OUTPUT (0x423), MINOR\_CREATE\_SSH\_LINK (0x42d), and  
MINOR\_CLOSE\_SSH\_LINK (0x42e);  
added 14 minor log types to the "MAJOR\_INFORMATION" log type:  
MINOR\_LIVE\_DETECT\_OPEN (0x400), MINOR\_LIVE\_DETECT\_CLOSE (0x401),  
MINOR\_CLEAR\_DATA\_COLLECTION (0x402), MINOR\_DELETE\_DATA\_COLLECTION (0x403),  
MINOR\_EXPORT\_DATA\_COLLECTION (0x404), MINOR\_CARD\_LEN\_CONFIG (0x405),  
MINOR\_DATA\_BASE\_INIT\_FAILED (0x406), MINOR\_DATA\_BASE\_PATCH\_UPDATE (0x407),  
MINOR\_PSAM\_CARD\_INSERT (0x408), MINOR\_PSAM\_CARD\_REMOVE (0x409),  
MINOR\_HARD\_FAULT\_REBOOT (0x40a), MINOR\_PSAM\_CARD\_OCP (0x40b),  
MINOR\_STACK\_OVERFLOW (0x40c), and MINOR\_PARM\_CFG (0x40d).

### Summary of Changes in Version 6.1.0.151\_July, 2019

1. Extended person management capability [\*\*\*JSON\\_Cap.UserInfo\*\*\*](#) (related API: [\*\*\*NET\\_DVR\\_STDXMLConfig\*\*\*](#); URL: [\*\*\*/ISAPI/AccessControl/UserInfo/capabilities?format=json\*\*\*](#)): added four nodes, i.e., "roomNumber" (room No.), "floorNumber" (floor No.), "callNumbers" (room No. list to be called), and "floorNumbers" (floor No. list).
2. Extended person information message [\*\*\*JSON\\_UserInfo\*\*\*](#) (related API: [\*\*\*NET\\_DVR\\_StartRemoteConfig\*\*\*](#); URLs: [\*\*\*/ISAPI/AccessControl/UserInfo/Record?format=json\*\*\*](#), [\*\*\*/ISAPI/AccessControl/UserInfo/Modify?format=json\*\*\*](#), and [\*\*\*/ISAPI/AccessControl/UserInfo/SetUp?format=json\*\*\*](#)):

- added two nodes, i.e., "**callNumbers**" (room No. list to be called) and "**floorNumbers**" (floor No. list).
3. Extended access control capability **XML\_AcsAbility** (related API: **NET\_DVR\_GetDeviceAbility** ; capability type: 0x801-"ACS\_ABILITY"):  
added two sub nodes, i.e., <**outdoorModules**> (whether to support upgrading modules of door station) and <**modules**> (supported module types) to the node <**AcsUpgrade**>.

## Summary of Changes in Version 6.1.0.11\_July, 2019

1. Extended person management capability message **JSON\_Cap\_UserInfo** and person information message **JSON\_UserInfo** (related URLs: </ISAPI/AccessControl/UserInfo/capabilities?format=json> , </ISAPI/AccessControl/UserInfo/Record?format=json> , and </ISAPI/AccessControl/UserInfo/Modify?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node **addUser** (whether to add the person if the person information being edited does not exist);  
added a person authentication mode "cardOrFpOrPw" (card or fingerprint or password) to the node **userVerifyMode**.
2. Extended person information search result message **JSON\_UserInfoSearch** (related URL: </ISAPI/AccessControl/UserInfo/Search?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a person authentication mode "cardOrFpOrPw" (card or fingerprint or password) to the sub node **userVerifyMode** of the node **UserInfo** (person information).
3. Extended card information capability message **JSON\_Cap\_CardInfo** and card information message **JSON\_CardInfo** (related URLs: </ISAPI/AccessControl/CardInfo/capabilities?format=json> and </ISAPI/AccessControl/CardInfo/Modify?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node **addCard** (whether to add the card if the card information being edited does not exist).
4. Extended condition message of searching for access control events **JSON\_AcsEventCond** (related URL: </ISAPI/AccessControl/AcsEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a node **eventAttribute** (event attribute).
5. Extended result message of searching for access control events **JSON\_AcsEvent** (related URL: </ISAPI/AccessControl/AcsEvent?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added two sub nodes: **attendanceStatus** (attendance status) and **statusValue** (status value) to the node **InfoList** (event details);  
added an authentication mode "cardOrFpOrPw" (card or fingerprint or password) to the sub node **currentVerifyMode** of the node **InfoList** (event details).
6. Extended capability message of searching for access control events **JSON\_Cap\_AcsEvent** (related URL: </ISAPI/AccessControl/AcsEvent/capabilities?format=json> ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a sub node **eventAttribute** (event attribute) to the node **AcsEventCond** (search conditions);  
added two sub nodes: **attendanceStatus** (attendance status) and **statusValue** (status value) to the node **InfoList** (event details);

- added an authentication mode "cardOrFpOrPw" (card or fingerprint or password) to the sub node **currentVerifyMode** of the node **InfoList** (event details).
7. Extended condition message of getting the total number of access control events by conditions **JSON AcsEventTotalNumCond** (related URL: [/ISAPI/AccessControl/AcsEventTotalNum?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a node **eventAttribute** (event attribute).
8. Extended capability message of getting the total number of the access control events by conditions **JSON Cap\_AcsEventTotalNum** (related URL: [/ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added a sub node **eventAttribute** (event attribute) to the node **AcsEventTotalNumCond** (search conditions).
9. Extended access control capability message **XML\_Cap\_AccessControl** (related URL: [/ISAPI/AccessControl/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added five nodes: <isSupportRemoteControlPWChcek> (whether to support verifying the password for remote door control), <isSupportRemoteControlPWCfg> (whether to support configuring password for remote door control), <isSupportAttendanceStatusModeCfg> (whether to support configuring attendance mode), <isSupportAttendanceStatusRuleCfg> (whether to support configuring attendance status and rule), and <isSupportCaptureCardInfo> (whether to support collecting card information).
10. Extended condition structure about getting access control events  
**NET\_DVRACS\_EVENT\_COND** (related API: [NET\\_DVR\\_StartRemoteConfig](#) ):  
added a member **byEventAttribute** (event attribute) by one reserved byte.
11. Extended access control event details structure **NET\_DVRACS\_EVENT\_DETAIL** (related API: [NET\\_DVR\\_StartRemoteConfig](#) ):  
added three members: **byAttendanceStatus** (attendance status), **byStatusValue** (attendance status value), and **byEventAttribute** (event attribute) by three reserved bytes;  
added an authentication mode 27 (card or fingerprint or password) to the member **byCurrentVerifyMode**.
12. Extended structure about extended access control event information  
**NET\_DVRACS\_EVENT\_INFO\_EXTEND** :  
added three members: **byAttendanceStatus** (attendance status), **byStatusValue** (attendance status value), and **byUUID** (UUID) by 38 bytes.
13. Added the function of configuring attendance status, refer to [Configure Attendance Status and Schedule](#) for details.
14. Added the function of collecting card information, refer to [Collect Card Information](#) for details.
15. Extended access control capability message **XML\_AcsAbility** (related API:  
[NET\\_DVR\\_GetDeviceAbility](#) ; capability type: 0x801-"ACS\_ABILITY"):  
added a sub node <ubootUpgrade> (whether to support upgrading uboot) to the node <AcsUpgrade>;  
added an authentication mode "cardOrFpOrPw" (card or fingerprint or password) to the sub node <verifyType> of the node <WeekPlan> of <DoorStatusPlan> and <CardReaderVerifyTypePlan>, respectively;

added an authentication mode 27 (card or fingerprint or password) to the sub node `<defaultVerifyMode>` of the node `<CardReaderCfg>`.

## 16. Extended the access control event types in [Access Control Event Types](#) :

added six event types to MAJOR\_EVENT: "MINOR\_LOCAL\_UPGRADE\_FAIL" (Local Upgrade Failed), "MINOR\_REMOTE\_UPGRADE\_FAIL" (Remote Upgrade Failed), "MINOR\_REMOTE\_EXTEND\_MODULE\_UPGRADE\_SUCC" (Extension Module is Remotely Upgraded), "MINOR\_REMOTE\_EXTEND\_MODULE\_UPGRADE\_FAIL" (Upgrading Extension Module Remotely Failed), "MINOR\_REMOTE\_FINGER\_PRINT\_MODULE\_UPGRADE\_SUCC" (Fingerprint Module is Remotely Upgraded), and "MINOR\_REMOTE\_FINGER\_PRINT\_MODULE\_UPGRADE\_FAIL" (Upgrading Fingerprint Module Remotely Failed).

## Summary of Changes in Version 6.1.0.11\_June, 2019

1. Extended configuration capability message of intelligent identity recognition terminal [XML\\_Cap\\_IdentityTerminal](#) and parameter message of intelligent identity recognition terminal [XML\\_IdentityTerminal](#) (related URLs: [/ISAPI/AccessControl/IdentityTerminal/capabilities](#) and [/ISAPI/AccessControl/IdentityTerminal](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added a node `<readCardRule>` (card No. setting rule).
2. Added the function of managing face information (including creating face picture library, managing face records in the face picture library, and configuring facial recognition mode), refer to [Manage Face Information](#) for details.
3. Extended person management capability message [JSON\\_Cap\\_UserInfo](#) (related URL: [/ISAPI/AccessControl/UserInfo/capabilities?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added a sub node `searchID` (search ID) to the node `UserInfoSearchCond` (search conditions);  
added a node `maxRecordNum` (supported maximum number of records (person records)).
4. Extended card information capability message [JSON\\_Cap\\_CardInfo](#) (related URL: [/ISAPI/AccessControl/CardInfo/capabilities?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added a sub node `searchID` (search ID) to the node `CardInfoSearchCond` (search conditions);  
added a node `maxRecordNum` (supported maximum number of records (card records)).
5. Extended fingerprint configuration capability message [JSON\\_Cap\\_FingerPrintCfg](#) (related URL: [/ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added a node `searchID` (search ID).
6. Added the function of getting information of face modeling failure after upgrading device, refer to [Other Configurations](#) for details.
7. Extended access control capability message [XML\\_Cap\\_AccessControl](#) (related URL: [/ISAPI/AccessControl/capabilities](#); related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added three nodes: `<isSupportCaptureFace>` (whether to support collecting face pictures), `<isSupportCaptureInfraredFace>` (whether to support collecting infrared face pictures), and `<isSupportFaceRecognizeMode>` (whether to support configuring facial recognition mode).

## Summary of Changes in Version 6.1.0.10\_July, 2019

1. Added the function of enabling or disabling NFC (Near-Field Communication) function (related API: [NET\\_DVR\\_STDXMLConfig](#)):  
Get the configuration capability: GET [/ISAPI/AccessControl/Configuration/NFCCfg/capabilities?format=json](#) ;  
Get parameters: GET [/ISAPI/AccessControl/Configuration/NFCCfg?format=json](#) ;  
Set parameters: PUT [/ISAPI/AccessControl/Configuration/NFCCfg?format=json](#) .
2. Added the function of enabling or disabling RF (Radio Frequency) card recognition (related API: [NET\\_DVR\\_STDXMLConfig](#)):  
Get the configuration capability: GET [/ISAPI/AccessControl/Configuration/RFCardCfg/capabilities?format=json](#) ;  
Get parameters: GET [/ISAPI/AccessControl/Configuration/RFCardCfg?format=json](#) ;  
Set parameters: PUT [/ISAPI/AccessControl/Configuration/RFCardCfg?format=json](#) .
3. Extended access control capability message [XML\\_Cap\\_AccessControl](#) (related URL: [/ISAPI/AccessControl/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#)):  
added two nodes: <isSupportNFCCfg> (whether the device supports enabling or disabling NFC function ) and <isSupportRFCardCfg> (whether the device supports enabling or disabling RF card recognition).
4. Extended access control capability message [XML\\_AcsAbility](#) (related API: [NET\\_DVR\\_GetDeviceAbility](#) ; capability type: "0x801-ACS\_ABILITY"):  
added eight event types to the sub node <EventEntry> (index: 3) of the node <EventLinkage> (event card linkage): "InformalMifareCardVerifyFail" (authentication failed: invalid Mifare card), "CPUCardEncryptVerifyFail" (verifying CPU card encryption failed), "NFCDisableVerifyFail" (disabling NFC verification failed), "EMCardRecognizeNotEnabled" (EM card recognition is disabled), "M1CardRecognizeNotEnabled" (M1 card recognition is disabled), "CPUCardRecognizeNotEnabled" (CPU card recognition is disabled), "IDCardRecognizeNotEnabled" (ID card recognition is disabled), and "CardSetSecretKeyFail" (importing key to the card failed).
5. Extended the access control event types in [Access Control Event Types](#) :  
added four operation event types to MAJOR\_OPERATION:  
"MINOR\_M1\_CARD\_ENCRYPT\_VERIFY\_OPEN" (M1 Card Encryption Verification Enabled),  
"MINOR\_M1\_CARD\_ENCRYPT\_VERIFY\_CLOSE" (M1 Card Encryption Verification Disabled),  
"MINOR\_NFC\_FUNCTION\_OPEN" (Opening Door with NFC Card Enabled), and  
"MINOR\_NFC\_FUNCTION\_CLOSE" (Opening Door with NFC Card Disabled);  
added eight event types to MAJOR\_EVENT: "MINOR\_INFORMAL\_MIFARE\_CARD\_VERIFY\_FAIL" (Authentication Failed: Invalid Mifare Card), "MINOR\_CPU\_CARD\_ENCRYPT\_VERIFY\_FAIL" (Verifying CPU Card Encryption Failed), "MINOR\_NFC\_DISABLE\_VERIFY\_FAIL" (Disabling NFC Verification Failed ), "MINOR\_EM\_CARD\_RECOGNIZE\_NOT\_ENABLED" (EM Card Recognition Disabled), "MINOR\_M1\_CARD\_RECOGNIZE\_NOT\_ENABLED" (M1 Card Recognition Disabled), "MINOR\_CPU\_CARD\_RECOGNIZE\_NOT\_ENABLED" (CPU Card Recognition Disabled),

"MINOR\_ID\_CARD\_RECOGNIZE\_NOT\_ENABLED" (ID Card Recognition Disabled), and  
"MINOR\_CARD\_SET\_SECRET\_KEY\_FAIL" (Importing Key to Card Failed).

## 6. Extended the event linkage types in *Event Linkage Types* :

added eight event linkage types of the authentication unit:

"EVENT\_ACS\_INFORMAL\_MIFARE\_CARD\_VERIFY\_FAIL" (Authentication Failed: Invalid Mifare Card), "EVENT\_ACS\_CPU\_CARD\_ENCRYPT\_VERIFY\_FAIL" (Verifying CPU Card Encryption Failed), "EVENT\_ACS\_NFC\_DISABLE\_VERIFY\_FAIL" (Disabling NFC Verification Failed ), "EVENT\_ACS\_EM\_CARD\_RECOGNIZE\_NOT\_ENABLED" (EM Card Recognition Disabled), "EVENT\_ACS\_M1\_CARD\_RECOGNIZE\_NOT\_ENABLED" (M1 Card Recognition Disabled), "EVENT\_ACS\_CPU\_CARD\_RECOGNIZE\_NOT\_ENABLED" (CPU Card Recognition Disabled), "EVENT\_ACS\_ID\_CARD\_RECOGNIZE\_NOT\_ENABLED" (ID Card Recognition Disabled), and "EVENT\_ACS\_CARD\_SET\_SECRET\_KEY\_FAIL" (Importing Key to Card Failed).

## Summary of Changes in Version 6.0.2.25\_May, 2019

1. Extended person information message **JSON UserInfo** and person information search result message **JSON UserInfoSearch** (related URLs: [/ISAPI/AccessControl/UserInfo/Record?format=json](#) , [/ISAPI/AccessControl/UserInfo/Search?format=json](#) , and [/ISAPI/AccessControl/UserInfo/Modify?format=json](#) ; related API: **NET\_DVR\_STDXMLConfig** ):  
added 11 person authentication modes "faceOrFpOrCardOrPw" (face or fingerprint or card or password), "faceAndFp" (face+fingerprint), "faceAndPw" (face+password), "faceAndCard" (face+card), "face" (face), "faceAndFpAndCard" (face+fingerprint+card), "faceAndPwAndFp" (face+password+fingerprint), "employeeNoAndFace" (employee No.+face), "faceOrfaceandCard" (face or face+card), "fpOrface" (fingerprint or face), "cardOrfaceOrPw" (card or face or password) to the sub node **userVerifyMode** in the node **UserInfo**.
2. Added the function of setting person information (related API: **NET\_DVR\_STDXMLConfig** ):  
**PUT** [/ISAPI/AccessControl/UserInfo/SetUp?format=json](#) .
3. Extended person management capability message **JSON Cap UserInfo** (related URL: [/ISAPI/AccessControl/UserInfo/capabilities?format=json](#) ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a function type "setUp" (set person information) to the node **supportFunction**;  
added two sub nodes **timeRangeBegin** (start time that can be configured) and **timeRangeEnd** (end time that can be configured) to the node **Valid**.
4. Extended card information message **JSON CardInfo** (related URL: [/ISAPI/AccessControl/CardInfo/Record?format=json](#) ; related API: **NET\_DVR\_STDXMLConfig** ):  
add a node **checkEmployeeNo** (whether to check the existence of the employee No. (person ID)).
5. Added the function of setting card information (related API: **NET\_DVR\_STDXMLConfig** ): **PUT** [/ISAPI/AccessControl/CardInfo/SetUp?format=json](#) .
6. Extended card information capability message **JSON Cap CardInfo** (related URL: [/ISAPI/AccessControl/CardInfo/capabilities?format=json](#) ; related API: **NET\_DVR\_STDXMLConfig** ):  
added a function type "setUp" (set card information) to the node **supportFunction**;  
added a node **checkEmployeeNo** (whether to check the existence of the employee No. (person ID)).

7. Added the function of managing fingerprint, refer to [\*\*Manage Fingerprint Information\*\*](#) for details.
8. Extended parameter union about event and card linkage configuration  
**NET\_DVR\_EVENTN\_CARD\_LINKAGE\_UNION** (related API: [\*\*NET\\_DVR\\_SetDeviceConfig\*\*](#)):  
added a member **byEmployeeNo** (employee No. (person ID)) by 32 bytes.
9. Extended parameter structure about event and card linkage configuration  
**NET\_DVR\_EVENT\_CARD\_LINKAGE\_CFG\_V51** (related API: [\*\*NET\\_DVR\\_SetDeviceConfig\*\*](#)):  
added a linkage type "3" (employee No. (person ID) linkage) to the member **byProMode**.
10. Extended access control event details structure **NET\_DVR\_ACS\_EVENT\_DETAIL** (related API: [\*\*NET\\_DVR\\_StartRemoteConfig\*\*](#)):  
added 11 authentication modes: 10 (face or fingerprint or card or password), 11 (face +fingerprint), 12 (face+password), 13 (face+card), 14 (face), 19 (face+fingerprint+card), 20 (face +password+fingerprint), 21 (employee No.+face), 22 (face or face+card), 23 (fingerprint or face), and 24 (card or face or password) to the member **byCurrentVerifyMode**;  
added a member **byEmployeeNo** (employee No. (person ID)).
11. Added the function of getting the total number of access control events by specific conditions (related API: [\*\*NET\\_DVR\\_STDXMLConfig\*\*](#)):  
Get the capability: GET [\*\*/ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json\*\*](#) ;  
Get the total number: POST [\*\*/ISAPI/AccessControl/AcsEventTotalNum?format=json\*\*](#) .
12. Added the structure about extended access control event information  
**NET\_DVR\_ACS\_EVENT\_INFO\_EXTEND** .
13. Extended access control alarm/event information structure **NET\_DVR\_ACS\_ALARM\_INFO** :  
added two members **pAcsEventInfoExtend** (it points to the structure **NET\_DVR\_ACS\_EVENT\_INFO\_EXTEND** when it is set to 1) and **byAcsEventInfoExtend** (whether **pAcsEventInfoExtend** is valid).
14. Added the function of log mode configuration and event optimization configuration, refer to [\*\*Other Configurations\*\*](#) for details.
15. Extended fingerprint and card reader parameters structure  
**NET\_DVR\_CARD\_READER\_CFG\_V50** (related API: [\*\*NET\\_DVR\\_SetDVRConfig\*\*](#)):  
added two members **bySupportDelFPByID** (whether the fingerprint and card reader supports deleting fingerprint by finger ID) and **byDefaultVerifyMode** (default authentication mode of the fingerprint and card reader (factory settings)).
16. Extended access controller configuration parameter structure **NET\_DVR\_ACS\_CFG** (related API: [\*\*NET\\_DVR\\_GetDVRConfig\*\*](#)):  
added a member **byProtocol** (communication protocol type of the card reader).
17. Extended expiry date configuration structure **NET\_DVR\_VALID\_PERIOD\_CFG** (related API: [\*\*NET\\_DVR\\_GetDVRConfig\*\*](#) and [\*\*NET\\_DVR\\_SetDVRConfig\*\*](#)):  
added a member **byTimeType** (time type).
18. Added the function of setting OSDP (Open Supervised Device Protocol) card reader, refer to [\*\*Device Settings\*\*](#) for details.
19. Extended access control capability message **XML\_AcsAbility** (related API: [\*\*NET\\_DVR\\_GetDeviceAbility\*\*](#) ; capability type: "0x801-ACS\_ABILITY"):

- added two sub nodes <**timeRangeBegin**> (start time that can be configured) and <**timeRangeEnd**> (end time that can be configured) to the node <**Card**> (card parameters capability);  
added a sub node <**supportDelFPByID**> (whether the fingerprint module supports deleting fingerprint by finger ID) to the node <**CardReaderCfg**> (reader parameters capability);  
added a sub node <**cardReaderFPAAlgorithmUpgrade**> (whether to enable upgrading fingerprint algorithm program of the fingerprint module) to the node <**AcsUpgrade**> (access control upgrading capability);  
added six sub nodes <**isNotSupportOpenDoor**> (whether the opening door linkage is not supported), <**isNotSupportCloseDoor**> (whether the closing door linkage is not supported), <**isNotSupportNormalOpen**> (whether the remaining door open is not supported), <**isNotSupportNormalClose**> (whether the remaining door closed is not supported), <**isNotSupportAlarmout**> (whether the alarm output linkage is not supported), and <**isNotSupportCapturePic**> (whether the capture linkage is not supported) to the node <**EventLinkage**> (event card linkage);  
added an event type "LegalEventNearlyFull" (alarm of no memory for legal offline event storage) to the sub node <**EventEntry**> (index: 0) of the node <**EventLinkage**> (event card linkage);  
added a sub node <**isSupportFingerCover**> (whether to overwrite the old fingerprint information when applying a new fingerprint information linked to the same employee No. (person ID)) to the node <**FingerPrint**> (fingerprint parameters).  
20. Extended access control capability message [XML Cap AccessControl](#) (related URL: [/ISAPI/AccessControl/capabilities](#) ; related API: [NET\\_DVR\\_STDXMLConfig](#) ):  
added 49 nodes: from <**isSupportRemoteControlDoor**> to <**isSupportLogModeCfg**>.   
21. Added two error codes to [Device Network SDK Errors](#) :  
1925-NET\_ERR\_NOT\_SUPPORT\_DEL\_FP\_BY\_ID (the fingerprint module does not support deleting fingerprint by finger ID) and 1926-NET\_ERR\_TIME\_RANGE (invalid range of the effective period).  
22. Added a sub status code 0x60001024—"eventNotSupport" (event subscription is not supported) to status code 6 (Invalid Message Content) in [Response Codes of Text Protocol](#) .

### Summary of Changes in Version 5.3.6.0\_Sept., 2018

1. Added person management functional module, refer to for details.
2. Edited the integration method of card management and fingerprint management: passed through the ISAPI protocol via HCNetSDK API to realize the functions. See and for details.
3. Added the control of buzzer and elevator, refer to [Remotely Control Door, Elevator, and Buzzer](#) for details.
4. Edited the flow of access control event configuration and added the functions of receiving access control event in listening mode, refer to [Configure Access Control Event](#) and [Receive Alarm/Event in Listening Mode](#) for details.

### Summary of Changes in Version 5.3.2.10\_Feb., 2018

New document.

## Chapter 2 Typical Applications

### 2.1 Data Collection

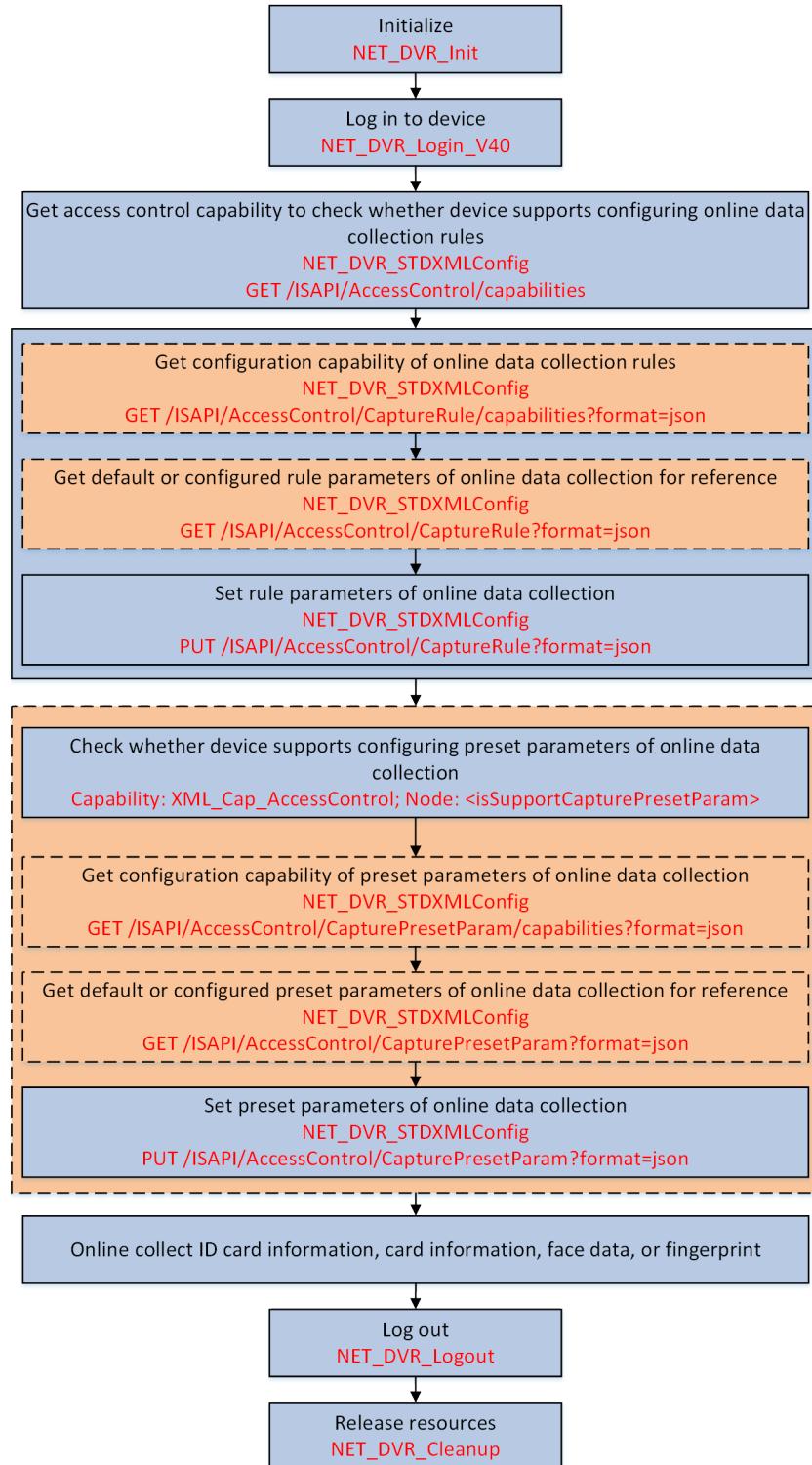
#### 2.1.1 Online Collect Data

When the access control device is connected to the client software or platform via the network, you can collect data (including ID card information, card information, face data, and fingerprint) on the client software or platform remotely. The online collected data will be uploaded to the client software or platform in real time.

##### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to the device.

## Steps



**Figure 1-1 Programming Flow of Online Collecting Data**

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/capabilities** for getting the access control capability to check whether the device supports configuring online data collection rules.

The access control capability is returned in **XML\_Cap\_AccessControl** by **IpOutputParam**.

If the device supports, the node <isSupportCaptureRule> is returned in the capability message and its value is "true", and then you can perform the following steps.

Otherwise, rule configuration of online data collection is not supported, please end this task.

2. Configure online data collection rules.

- 1) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CaptureRule/capabilities?format=json** for getting the configuration capability of online data collection rules.

The capability is returned in the message **JSON\_CaptureRuleCap** by **IpOutputParam**.

- 2) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CaptureRule?format=json** for getting default or configured rule parameters of online data collection for reference.

The rule parameters are returned in the message **JSON\_CaptureRule** by **IpOutputParam**.

- 3) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/CaptureRule?format=json** and set **IpInputParam** to **JSON\_CaptureRule** for setting rule parameters of online data collection.

3. **Optional:** Configure preset parameters of online data collection.

- 1) Check the access control capability **XML\_Cap\_AccessControl** to know whether the device supports configuring preset parameters of online data collection.

If the device supports, the node <isSupportCapturePresetParam> is in the capability message and its value is "true", and then you can continue to set preset parameters.

Otherwise, preset configuration of online data collection is not supported.

- 2) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CapturePresetParam/capabilities?format=json** for getting the configuration capability of preset parameters of online data collection.

The configuration capability is returned in the message **JSON\_CapturePresetCap** by **IpOutputParam**.

- 3) **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CapturePresetParam?format=json** for getting default or configured preset parameters of online data collection for reference.

The preset parameters are returned in the message **JSON\_CapturePreset** by **IpOutputParam**.

- 4) Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/CapturePresetParam?format=json** and set **IpInputParam** to the message **JSON\_CapturePreset** for setting preset parameters of online data collection.



## Note

The preset parameters are used to display custom information on the device UI during data collection. Currently, it only supports displaying the name of the person whose data is being collected. The preset parameters should be configured again for each collection.

4. Perform the following operation(s) to collect ID card information, card information, face data, or fingerprint online.

### Collect ID Card Information

- a. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET /[ISAPI/AccessControl/capabilities](#) for getting access control capability to check whether the device supports online collecting ID card information.  
The capability is returned in the message [XML\\_Cap\\_AccessControl](#) by **IpOutputParam**. If it supports, the node <isSupportCaptureIDInfo> is returned and its value is "true". Otherwise, online ID card collection is not supported.
- b. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET /[ISAPI/AccessControl/CaptureIDInfo/capabilities?format=json](#) for getting the capability of online collecting ID card information.  
The capability is returned in the message [JSON\\_IdentityInfoCap](#) by **IpOutputParam**.
- c. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: POST /[ISAPI/AccessControl/CaptureIDInfo?format=json](#) and set **IpInputParam** to the message [JSON\\_IdentityInfoCond](#) for online collecting ID card information.  
The online collected ID card information is returned in the message [JSON\\_IdentityInfo](#) by **IpOutputParam**.

### Collect Card Information

Refer to [Collect Card Information](#)

### Collect Face Data

Refer to [Collect Face Data](#)

### Collect Fingerprint

Refer to [Fingerprint Collection](#)

## What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out of the device and release the resources.

### 2.1.2 Offline Collect Data

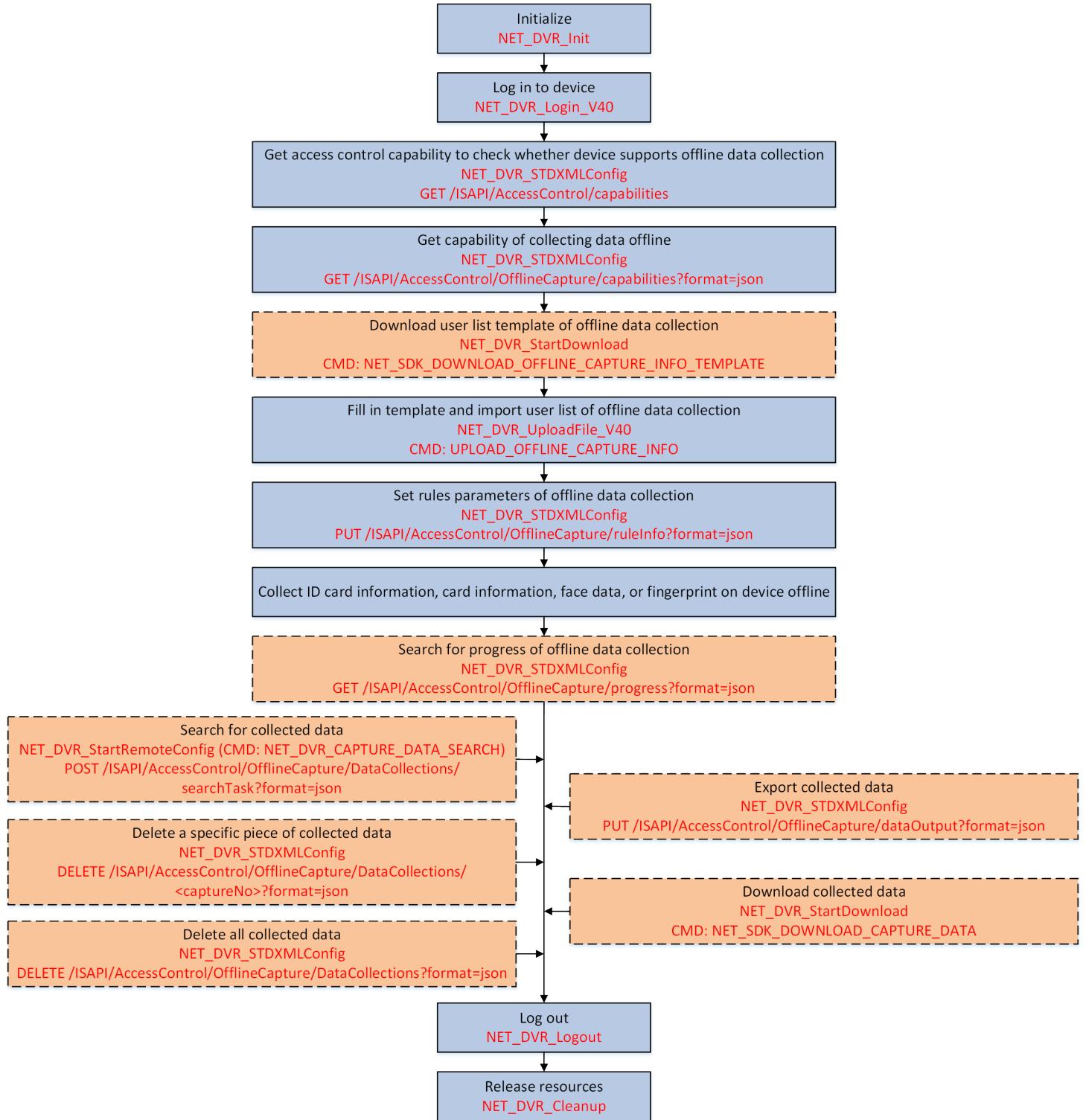
When the access control device is not connected to the client software or platform via the network, you can collect data (including ID card information, card information, face data, and fingerprint) locally on the stand-alone device by importing description of the information that

needs to be collected. The offline collected data will be stored on the device and can also be downloaded, exported, or deleted from the device.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps



**Figure 1-2 Programming Flow of Offline Collecting Data**

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/capabilities** for getting the access control capability to check whether the device supports offline data collection.

The capability is returned in the message **XML\_Cap\_AccessControl** by **IpOutBuffer** of **IpOutputParam**.

If this function is supported, the node <isSupportOfflineCapture> will be returned and its value is "true". Otherwise, please end this task.

2. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/OfflineCapture/capabilities?format=json** for getting the capability of collecting data offline to know the supported parameters.

The capability is returned in the message **JSON\_OfflineCaptureCap** **IpOutBuffer** of **IpOutputParam**.

3. **Optional:** Download the user list template of offline data collection.

- 1) Call **NET\_DVR\_StartDownload** and set the input parameter **dwDownloadType** to "NET\_SDK\_DOWNLOAD\_OFFLINE\_CAPTURE\_INFO\_TEMPLATE" (macro definition value: 40) to start downloading.
- 2) Call **NET\_DVR\_GetDownloadState** to get the downloading progress.
- 3) Call **NET\_DVR\_StopDownload** to stop downloading.

4. Import the user list of offline data collection filled in the template.

- 1) Call **NET\_DVR\_UploadFile\_V40**, set **dwUploadType** to "UPLOAD\_OFFLINE\_CAPTURE\_INFO" (macro definition value: 56), and set **IpInBuffer** to the structure **NET\_DVR\_DOOR\_FILE\_UPLOAD\_PARAM** for start uploading the file.
- 2) Call **NET\_DVR\_GetUploadState** to get the file uploading progress.
- 3) Call **NET\_DVR\_UploadClose** to stop uploading.



If importing failed, you can call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/OfflineCapture/uploadFailedDetails?format=json** for getting the details of failing to upload the user list of offline data collection.

The uploading failure details are returned in the message **JSON\_UploadFailedDetails** by **IpOutputParam**.

5. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json** and set **IpInBuffer** of **IpInputParam** to the message **JSON\_RuleInfo** for setting rule parameters of offline data collection.



Before setting rule parameters of offline data collection, you'd better call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json** for getting the existing or configured parameters for reference. The parameters are returned in the message **JSON\_RuleInfo** by **IpOutBuffer** of **IpOutputParam**.

6. Collect ID card information, card information, face data, or fingerprint on the stand-alone device offline.

7. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/OfflineCapture/progress?format=json](#) for getting the progress of offline data collection.

The collection progress is returned in the message [JSON\\_CaptureProgress](#) by **IpOutBuffer** of **IpOutputParam**.

8. **Optional:** Perform the following operation(s) after collecting data offline.

<b>Export Collected Data</b>	Call <a href="#"><u>NET_DVR_STDXMLConfig</u></a> to pass through the request URL: PUT <a href="#"><u>/ISAPI/AccessControl/OfflineCapture/dataOutput?format=json</u></a> and set <b>IpInBuffer</b> of <b>IpInputParam</b> to the message <a href="#"><u>JSON_DataOutputCfg</u></a> .
------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



#### Note

During exporting, you can call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/OfflineCapture/dataOutput/progress?format=json](#) for getting the progress of exporting the offline collected data.

<b>Download Collected Data</b>	a. Call <a href="#"><u>NET_DVR_StartDownload</u></a> and set the <b>dwDownloadType</b> to "NET_SDK_DOWNLOAD_CAPTURE_DATA" (macro definition value: 41) to start downloading. b. Call <a href="#"><u>NET_DVR_GetDownloadState</u></a> to get the downloading status. c. Call <a href="#"><u>NET_DVR_StopDownload</u></a> to stop downloading.
--------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Search for Collected Data</b>	a. Call <a href="#"><u>NET_DVR_StartRemoteConfig</u></a> with "NET_DVR_CAPTURE_DATA_SEARCH" (command No.: 2554) and set the <b>IpInBuffer</b> to the request URI POST <a href="#"><u>/ISAPI/AccessControl/OfflineCapture/DataCollections/searchTask?format=json</u></a> for setting up persistent connection and set callback function ( <a href="#"><u>fRemoteConfigCallback</u></a> ) for searching for the collected data. b. Call <a href="#"><u>NET_DVR_SendRemoteConfig</u></a> to send the search condition message <a href="#"><u>JSON_SearchTaskCond</u></a> via the persistent connection. The collected data is returned in the structure <a href="#"><u>NET_DVR_JSON_DATA_CFG</u></a> by the output buffer ( <b>IpBuffer</b> ) of the callback function.
----------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



#### Note

- The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA".
- After a search condition message [JSON\\_SearchTaskCond](#) is applied by calling [NET\\_DVR\\_SendRemoteConfig](#), the next piece of data can be searched only when [NET\\_DVR\\_JSON\\_DATA\\_CFG](#) is returned by the callback function [fRemoteConfigCallback](#).

c. Call <a href="#"><u>NET_DVR_StopRemoteConfig</u></a> to disconnect the persistent connection and finishing searching.
--------------------------------------------------------------------------------------------------------------------------

<b>Delete A Specific Piece of Collected Data</b>	Call <a href="#"><u>NET_DVR_STDXMLConfig</u></a> to pass through the request URL: DELETE / <a href="#"><u>ISAPI/AccessControl/OfflineCapture/DataCollections/&lt;captureNo&gt;?format=json</u></a> .
<b>Delete All Collected Data</b>	Call <a href="#"><u>NET_DVR_STDXMLConfig</u></a> to pass through the request URL: DELETE / <a href="#"><u>ISAPI/AccessControl/OfflineCapture/DataCollections?format=json</u></a> .

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out of the device and release the resources.

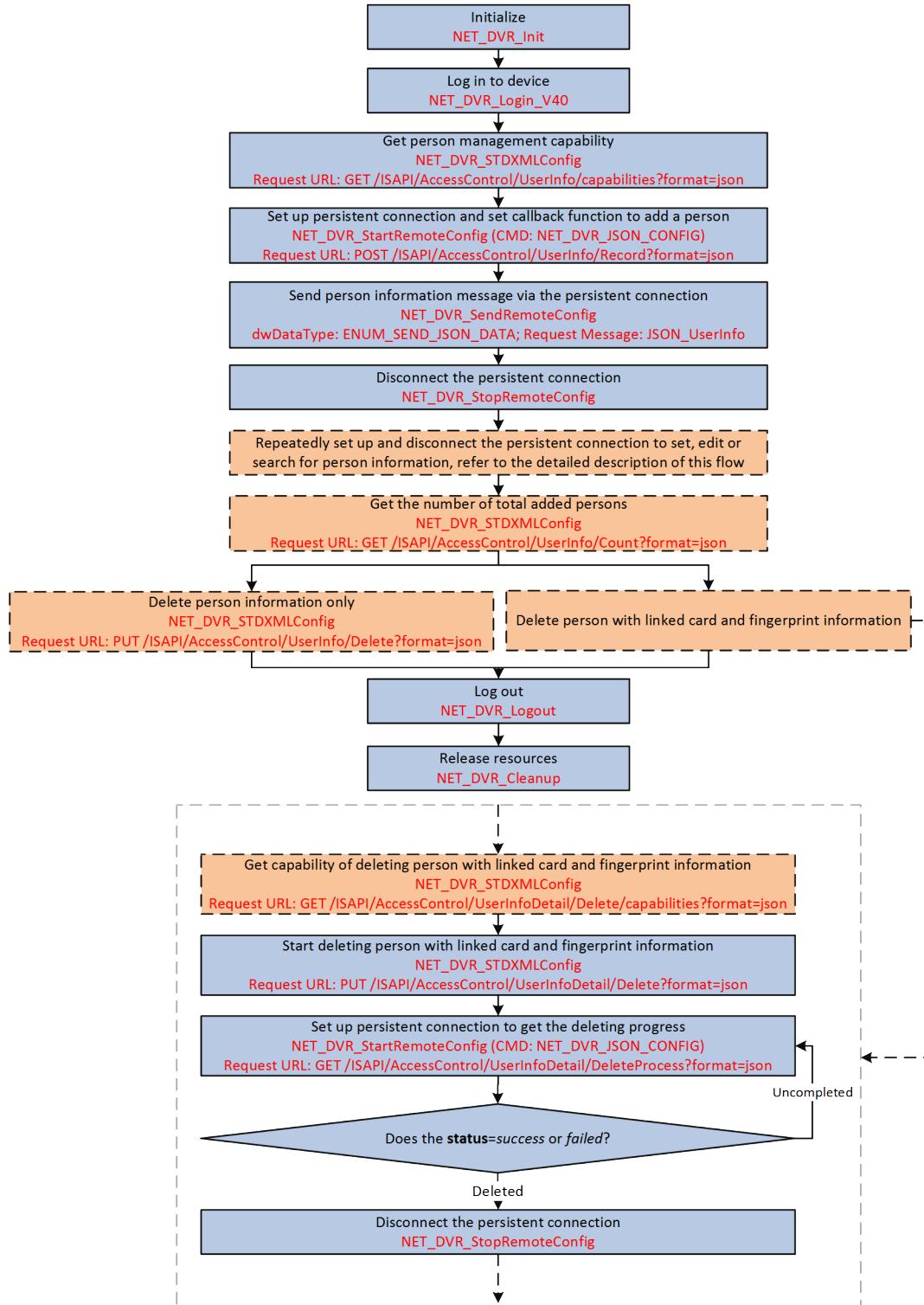
## 2.2 Manage Person Information

A person is a basic unit, which can link with multiple cards and fingerprints, for access control in this manual. So, before starting any other operations, you should add persons and apply the person information (e.g., person ID, name, organization, permissions, and so on) to access control devices.

### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to device.

## Steps



**Figure 1-3 Programming Flow of Managing Person Information**

1. Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/UserInfo/capabilities?format=json** for getting the person management capability to know the configuration details and notices.  
The capability message **JSON\_Cap\_UserInfo** is returned.
2. Call **NET\_DVR\_StartRemoteConfig** with the command "NET\_DVR\_JSON\_CONFIG" (command No.: 2550) to transmit the request URI: POST **/ISAPI/AccessControl/UserInfo/Record?format=json** for setting up persistent connection and set callback function for adding a person.
3. Call **NET\_DVR\_SendRemoteConfig** to send the person information message **JSON\_UserInfo** via the persistent connection.



The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA".

4. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish adding.
5. **Optional:** Perform the following operation(s) after adding persons.

## Set Person Information

- a. Call **NET\_DVR\_StartRemoteConfig** with the command "NET\_DVR\_JSON\_CONFIG" (command No.: 2550) to transmit the request URI: PUT **/ISAPI/AccessControl/UserInfo/SetUp?format=json** for setting up the persistent connection and set the callback function for setting person information.
- b. Call **NET\_DVR\_SendRemoteConfig** to send the person information message **JSON\_UserInfo** via the persistent connection.



The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA".

- c. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish setting.
- a. Call **NET\_DVR\_StartRemoteConfig** with the command "NET\_DVR\_JSON\_CONFIG" (command No.: 2550) to transmit the request URI: PUT **/ISAPI/AccessControl/UserInfo/Modify?format=json** for setting up persistent connection and set callback function for editing a person.
- b. Call **NET\_DVR\_SendRemoteConfig** to send the person information message **JSON\_UserInfo** via the persistent connection.



The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA".

- c. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish editing.
- a. Call **NET\_DVR\_StartRemoteConfig** with the command "NET\_DVR\_JSON\_CONFIG" (command No.: 2550) to transmit the request URI: POST **/ISAPI/AccessControl/UserInfo/Search?format=json** for

- setting up persistent connection and set callback function for searching for persons.
- b. Call ***NET\_DVR\_SendRemoteConfig*** to send the search condition message ***JSON\_UserInfoSearchCond*** via the persistent connection.
- 



#### Note

The type of data to be sent (***dwDataType***) should be set to "ENUM\_SEND\_JSON\_DATA", and the search results ***JSON\_UserInfoSearch*** will be returned in the callback function configured by the above step.

- c. Call ***NET\_DVR\_StopRemoteConfig*** to disconnect the persistent connection and finish searching.
- 

## Get Number of Total Added Persons

Call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/AccessControl/UserInfo/Count?format=json***.

---



#### Note

The number of total added persons will be returned in the message ***JSON\_UserInfoCount***.

---

## Delete Person Only

Call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: PUT ***/ISAPI/AccessControl/UserInfo/Delete?format=json*** and set ***lpInBuffer*** of ***lpInputParam*** to the message ***JSON\_UserInfoDelCond***.

---



#### Note

The timeout of deleting person only can be configured, and setting the timeout to 60s is suggested.

---

## Delete Person with Linked Card and Fingerprint

- a. Call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/AccessControl/UserInfoDetail/Delete/capabilities?format=json*** for getting capability of deleting person with linked card and fingerprint. And the capability is returned in the message ***JSON\_Cap\_UserInfoDetail*** by the output parameter (***lpOutputParam***).
- b. Call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: PUT ***/ISAPI/AccessControl/UserInfoDetail/Delete?format=json*** and set ***lpInBuffer*** of ***lpInputParam*** to the message ***JSON\_UserInfoDetail*** for starting deleting.
- c. Call ***NET\_DVR\_StartRemoteConfig*** with the command "NET\_DVR\_JSON\_CONFIG" (command No.: 2550) to transmit the request URI: GET ***/ISAPI/AccessControl/UserInfoDetail/DeleteProcess?format=json*** for setting up persistent connection to get the deleting progress.



## Note

If the value of node **status** in the deleting progress message **JSON UserInfoDetailDeleteProcess** is "success", it indicates that deleting completed and perform the next step; if the value is "failed", it refers to uncompleted deleting progress and you should repeatedly transmit the request URI: GET **/ISAPI/AccessControl/UserInfoDetail/DeleteProcess?format=json** to continuously getting the deleting progress.

---

- d. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish getting deleting progress.

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out off the device and release the resources.

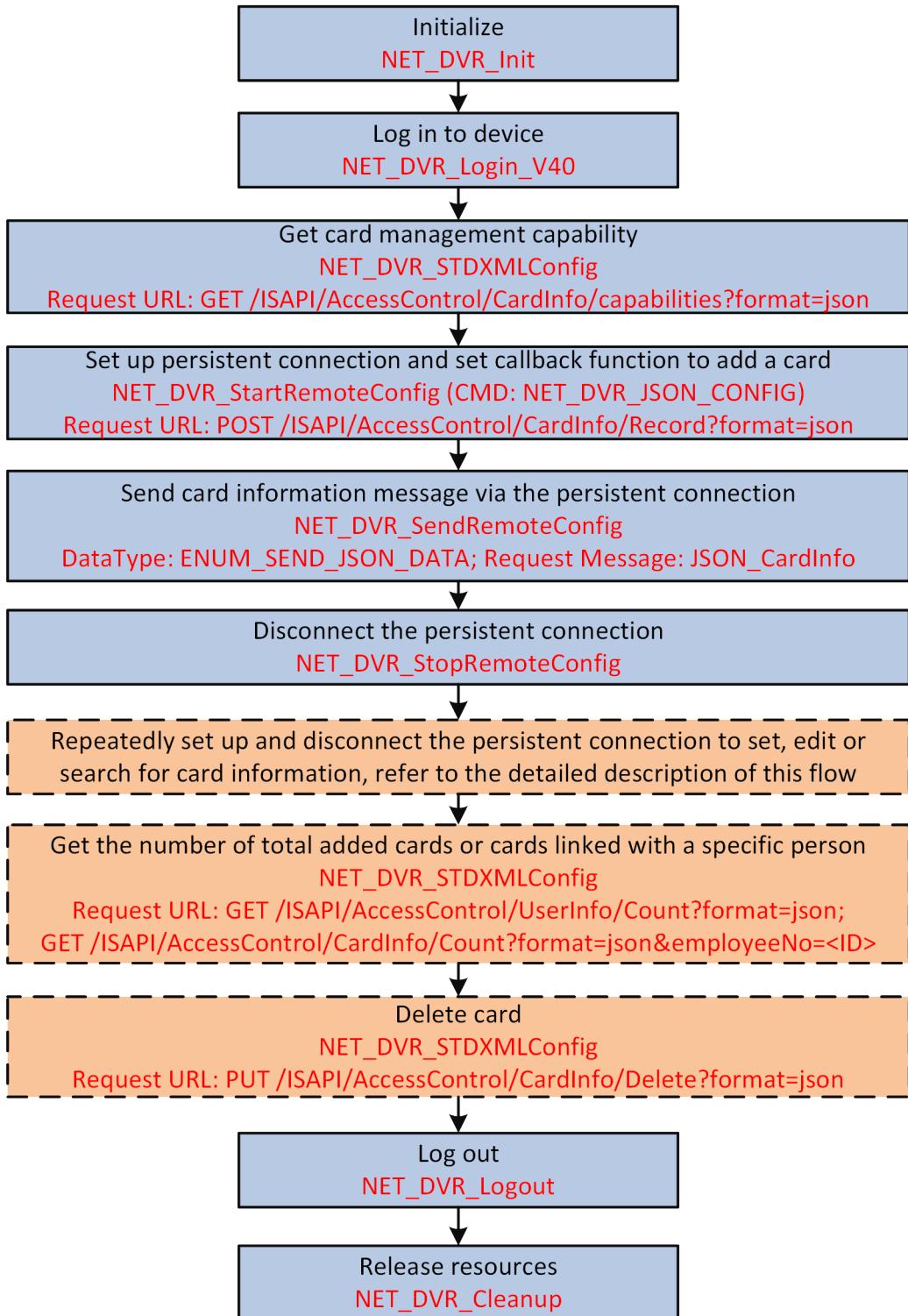
## 2.3 Manage Card Information

If a person want to access by card, you should add cards and link the cards with the person for getting the access permissions, and then apply card information (e.g., card No., card type, and so on) to access control device.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to device.
- Make sure you have collected the card information, refer to **Collect Card Information** for details.

## Steps



**Figure 1-4 Programming Flow of Managing Card Information**

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CardInfo/capabilities?format=json** for getting the card management capability to know the configuration details and notices.  
The capability message **JSON\_Cap\_CardInfo** is returned.
  2. Call **NET\_DVR\_StartRemoteConfig** with the command of NET\_DVR\_JSON\_CONFIG (command No.: 2550) to pass through the request URL: POST **/ISAPI/AccessControl/CardInfo/Record?format=json** for setting up persistent connection and set callback function for adding a card.
  3. Call **NET\_DVR\_SendRemoteConfig** to send the card information message **JSON\_CardInfo** via the persistent connection.
- 



The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA".

4. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finishing adding.
5. **Optional:** Perform the following operation(s) after adding cards.

<b>Set Card Information</b>	Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit the request URI: PUT <b><u>/ISAPI/AccessControl/CardInfo/SetUp?format=json</u></b> and set <b>lpInputParam</b> to the message <b><u>JSON_CardInfo</u></b>
<b>Edit Card Information</b>	<ol style="list-style-type: none"><li>a. Call <b><u>NET_DVR_StartRemoteConfig</u></b> with the command of NET_DVR_JSON_CONFIG (command No.: 2550) to pass through the request URL: PUT <b><u>/ISAPI/AccessControl/CardInfo/Modify?format=json</u></b> for setting up persistent connection and setting callback function to edit a card.</li><li>b. Call <b><u>NET_DVR_SendRemoteConfig</u></b> to send the card information message <b><u>JSON_CardInfo</u></b> via the persistent connection.</li></ol>
<b>Search for Cards</b>	<ol style="list-style-type: none"><li>c. Call <b><u>NET_DVR_StopRemoteConfig</u></b> to disconnect the persistent connection and finish editing.</li><li>a. Call <b><u>NET_DVR_StartRemoteConfig</u></b> with the command of NET_DVR_JSON_CONFIG (command No.: 2550) to pass through the request URL: POST <b><u>/ISAPI/AccessControl/CardInfo/Search?format=json</u></b> for setting up persistent connection and set callback function for searching for cards.</li><li>b. Call <b><u>NET_DVR_SendRemoteConfig</u></b> to send the search condition message <b><u>JSON_CardInfoSearchCond</u></b> via the persistent connection.</li></ol>



## Note

The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA", and the search results **JSON\_CardInfoSearch** will be returned in the callback function configured by the above step.

- c. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish searching.

### Get Number of Total Added Cards

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET [\*\*/ISAPI/AccessControl/CardInfo/Count?format=json\*\*](#).



## Note

The number of total added cards will be returned in the message **JSON\_CardInfoCount**.

### Get Number of Cards Linked with A Person

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET [\*\*/ISAPI/AccessControl/CardInfo/Count?format=json&employeeNo=<ID>\*\*](#).



## Note

The number of cards linked with a person will be returned in the message **JSON\_CardInfoCount**.

### Delete Card

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT [\*\*/ISAPI/AccessControl/CardInfo/Delete?format=json\*\*](#) and set **IplInBuffer** of **IplInputParam** to the message **JSON\_CardInfoDelCond**.



## Note

The timeout of deleting card can be configured, and setting the timeout to 60s is suggested.

## What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out off the device and release the resources.

### 2.3.1 Collect Card Information

The card information for further management and applying should be collected by the card reading module of the access control device first. The following contents are about the process and parameter settings of collecting card information.

#### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/capabilities** for getting access control capability to check whether the device supports card information collection.

The access control capability is returned in the message **XML\_Cap\_AccessControl** by **IpOutputParam**.

If the device supports card information collection, the node <isSupportCaptureCardInfo> will be returned and its value is "true", and then you can perform the following steps.

Otherwise, card information collection is not supported by the device, please end this task.

2. **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json** to get the capability of collecting card information.

The capability is returned in the message **JSON\_CardInfoCap** by **IpOutputParam**.

3. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CaptureCardInfo?format=json** to collect the card information.

The collected card information is returned in the message **JSON\_CardInfo\_Collection** by **IpOutputParam**.

## What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out of the device and release the resources.

## 2.3.2 Card Operation

### Get Card Operation Capability

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/CardOperations/capabilities?format=json** .

The capability is returned in the message **JSON\_CardOperationsCap** by **IpOutputParam**.

### Encrypt Specific Sections (M1 Card)

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/sectionEncryption?format=json** and set **IpInputParam** to **JSON\_SectionEncryption** .

### Verify Section Password (M1 Card)

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/verification?format=json** and set **IpInputParam** to **JSON\_Verification** .

### Change Control Block of Section (M1 Card)

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/controlBlock?format=json** and set **IpInputParam** to **JSON\_ControlBlock** .

### Read or Write Block Data (M1 Card)

#### Read Block Data

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/CardOperations/dataBlock/<address>?format=json**.

The block data is returned in the message **JSON\_DataBlock** by **IpOutputParam**.

## Write Block Data

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/dataBlock/<address>?format=json** and set **IpInputParam** to **JSON\_DataBlock**.

## Operate Data Block (M1 Card)

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/dataBlock/control?format=json** and set **IpInputParam** to **JSON\_DataBlockCtrl**.

## Set Operation Protocol Type of Card

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/protocol?format=json** and set **IpInputParam** to **JSON\_CardProto**.

## Set CPU Card Parameters

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/cardParam?format=json** and set **IpInputParam** to **JSON\_CardParam**.

## Reset CPU Card

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/CardOperations/reset?format=json**.

And the resetting result is returned in the message **JSON\_CardResetResponse** by **IpOutputParam**.

## Pass Through Data Package of CPU Card

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/dataTrans?format=json** and set **IpInputParam** to **JSON\_DataTrans**.

## Encrypt CPU Card

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/encryption?format=json** and set **IpInputParam** to **JSON\_CardEncryption**.

## Delete Data from Card

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/clearData?format=json** and set **IpInputParam** to **JSON\_ClearData**.

## Set Custom Card Information

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardOperations/customData?format=json** and set **IpInputParam** to **JSON\_CustomData**.

## Search for Custom Card Information

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: POST **/ISAPI/AccessControl/CardOperations/customData/searchTask?format=json** and set **IpInputParam** to **JSON\_CustomDataSearchCond**.

## 2.4 Manage Fingerprint Information

If a person wants to access by fingerprint, you should collect the fingerprint data via the fingerprint recorder first, and then apply the fingerprint data and parameters (e.g., fingerprint ID, type, and so on) to the fingerprint module of the access control device and link the fingerprint with the person for getting access permissions.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

### Steps

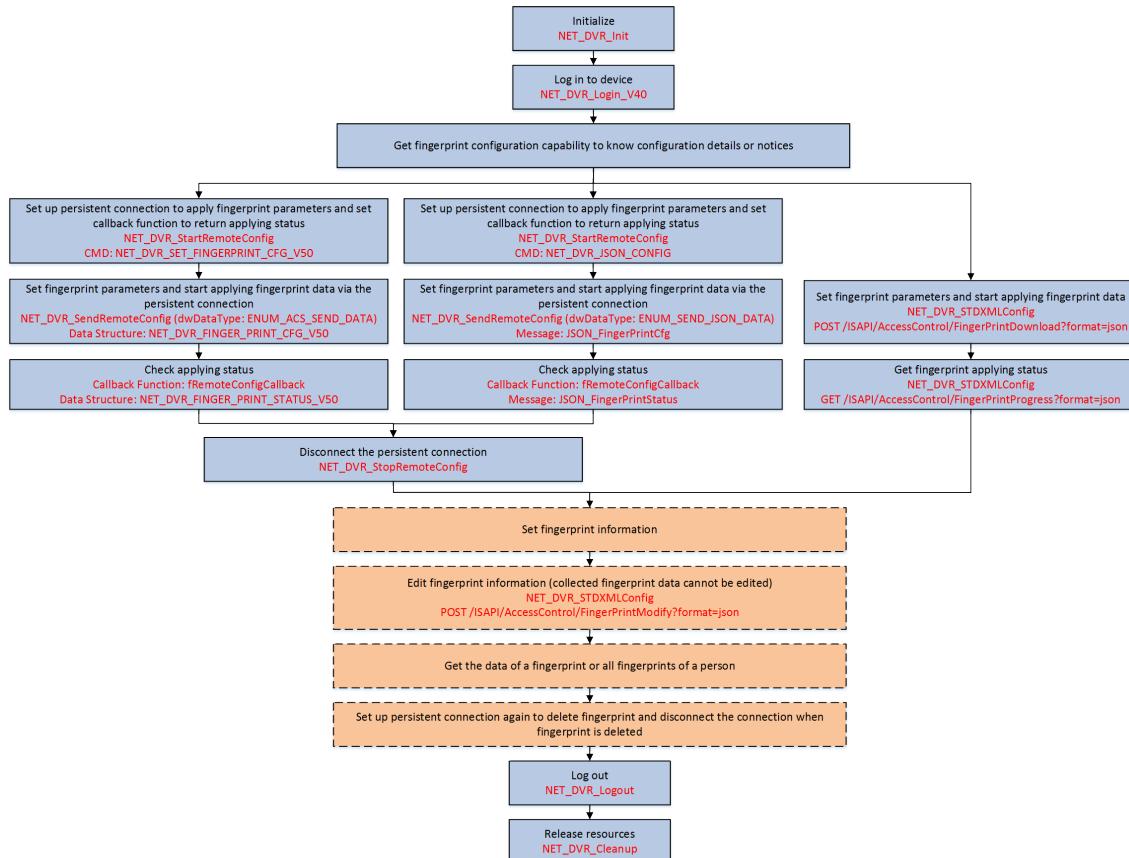


Figure 1-5 Programming Flow of Managing Fingerprint Information



To collect the fingerprint, refer to [\*\*Fingerprint Collection\*\*](#) for details.

---

## 1. Get the fingerprint configuration capability to know the configuration details or notices.

- Call [\*\*NET\\_DVR\\_GetDeviceAbility\*\*](#), set the capability type **dwAbilityType** to "ACS\_ABILITY", and set the input parameter pointer **pInBuf** to the message [\*\*XML\\_Desc\\_AcsAbility\*\*](#) for getting the fingerprint configuration and deleting capability to know the configuration details or notices.

The capability is returned in the message [\*\*XML\\_AcsAbility\*\*](#) by the output parameter pointer **pOutBuf**. The related nodes are <FingerPrint> and <DelFingerPrint>.

- Call [\*\*NET\\_DVR\\_STDXMLConfig\*\*](#) to transmit the request URI: GET [\*\*/ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json\*\*](#) for getting the fingerprint configuration capability to know the configuration details or notices.

The configuration capability is returned in the message [\*\*JSON\\_Cap\\_FingerPrintCfg\*\*](#) by the output parameter **IpOutputParam**.

## 2. Apply fingerprint data via the persistent connection.

- a. Call [\*\*NET\\_DVR\\_StartRemoteConfig\*\*](#) with "NET\_DVR\_SET\_FINGERPRINT\_CFG\_V50" (command No.: 2184) and set the input parameter **IpInBuffer** to the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_CFG\\_V50\*\*](#) for setting up the persistent connection, and set the callback function ([\*\*fRemoteConfigCallback\*\*](#)) for applying fingerprint information (i.e., fingerprint data and parameters) and returning the applying status.



- Before setting fingerprint parameters, you'd better call [\*\*NET\\_DVR\\_StartRemoteConfig\*\*](#) with "NET\_DVR\_GET\_FINGERPRINT\_CFG\_V50" (command No.: 2183) and set the input parameter **IpInBuffer** to the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_INFO\\_COND\\_V50\*\*](#) for getting the existing or configured parameters for reference, and the parameters are returned in the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_CFG\\_V50\*\*](#) by the output buffer (**IpBuffer**) of the callback function [\*\*fRemoteConfigCallback\*\*](#).

- When applying the fingerprint information, whether to apply the card No. or employee ID for linking with the fingerprint is determined by the capability [\*\*XML\\_AcsAbility\*\*](#). If the node <employeeNo> is returned in the capability, it indicates that the device supports applying fingerprint information based on person, so the card No. in the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_INFO\\_COND\\_V50\*\*](#) is not required, and the fingerprint will link with the person directly after applying.

- b. Call [\*\*NET\\_DVR\\_SendRemoteConfig\*\*](#), set the data type **dwDataType** to "ENUM\_AC\_SND\_DATA" (macro definition value: 0x3), and set the **pSendBuf** to the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_CFG\\_V50\*\*](#) for sending the fingerprint information via the persistent connection to start applying.
- c. Check the applying status in the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_STATUS\\_V50\*\*](#) returned by the output buffer (**IpBuffer**) of the callback function [\*\*fRemoteConfigCallback\*\*](#) of [\*\*NET\\_DVR\\_StartRemoteConfig\*\*](#).



### Note

When the parameter **byTotalStatus** in the structure is "1", it indicates that the fingerprint information is applied.

- d. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish setting and applying.
  - a. Call **NET\_DVR\_StartRemoteConfig** with "NET\_DVR\_JSON\_CONFIG" (command No.: 2550) and set the input parameter **lpInBuffer** to the request URI: POST **/ISAPI/AccessControl/FingerPrintDownload?format=json** for setting up the persistent connection, and set the callback function (**fRemoteConfigCallback**) for applying fingerprint information (i.e., fingerprint data and parameters) and returning the applying status.
  - b. Call **NET\_DVR\_SendRemoteConfig**, set the data type **dwDataType** to "ENUM\_SEND\_JSON\_DATA" (macro definition value: 0x3), and set the **pSendBuf** to the message **JSON\_FingerPrintCfg** for sending the fingerprint information via the persistent connection to start applying.
  - c. Check the applying status in the message **JSON\_FingerPrintStatus** returned by the output buffer (**lpBuffer**) of the callback function **fRemoteConfigCallback** of **NET\_DVR\_StartRemoteConfig**.
- 



### Note

When the parameter **totalStatus** in the message is "1", it indicates that the fingerprint information is applied.

- d. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish setting and applying.
  - a. Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: POST **/ISAPI/AccessControl/FingerPrintDownload?format=json** and set input parameter **lpInputParam** to the message **JSON\_FingerPrintCfg** and binary fingerprint data for setting the fingerprint parameters (e.g., employee No. to be linked, fingerprint modules to be applied, and so on) and starting applying the recorded fingerprint data.
- 



### Note

The binary fingerprint data is collected and recorded by the fingerprint recorder.

- b. Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/FingerPrintProgress?format=json** for getting the applying status to make sure that the applying is completed.
- 



### Note

The fingerprint data is linked to a person according to the configured employee No. and applied to the specified fingerprint modules only when the value of applying status (**totalStatus**) is "1".

- 3. **Optional:** Perform the following operation(s) after setting and applying fingerprint information.

<b>Set Fingerprint Information</b>	<p>Method 1:</p> <ol style="list-style-type: none"><li>Call <b><i>NET_DVR_StartRemoteConfig</i></b> with "NET_DVR_JSON_CONFIG" (command No.: 2550) and set the input parameter <b>lpInBuffer</b> to the request URI: POST <b>/ISAPI/AccessControl/FingerPrint/SetUp?format=json</b> for setting up persistent connection, and set the callback function <b>fRemoteConfigCallback</b> for setting the fingerprint information.</li><li>Call <b><i>NET_DVR_SendRemoteConfig</i></b>, set the data type <b>dwDataType</b> to "ENUM_SEND_JSON_DATA", and set the <b>pSendBuf</b> to the message <b>JSON_FingerPrintCfg</b> for sending the fingerprint information via the persistent connection.</li><li>Check the setting status in the message <b>JSON_FingerPrintStatus</b> returned by the output buffer (<b>lpBuffer</b>) of the callback function <b>fRemoteConfigCallback</b> of <b><i>NET_DVR_StartRemoteConfig</i></b>.</li><li>Call <b><i>NET_DVR_StopRemoteConfig</i></b> to disconnect the persistent connection and finish setting.</li></ol> <p>Method 2:</p> <p>Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: POST <b>/ISAPI/AccessControl/FingerPrint/SetUp?format=json</b> and set the input parameter <b>lpInputParam</b> to the message <b>JSON_FingerPrintCfg</b>.</p>
<b>Edit Fingerprint Information</b>	<p>Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: POST <b>/ISAPI/AccessControl/FingerPrintModify?format=json</b> and set the input parameter <b>lpInputParam</b> to the message <b>JSON_FingerPrintModify</b>.</p>
<b>Get Fingerprint Data</b>	<p> <b>Note</b></p> <p>Only the fingerprint parameters (such as fingerprint ID, type, and so on) can be edited. The collected fingerprint data cannot be edited.</p> <hr/> <p>Method 1:</p> <ol style="list-style-type: none"><li>Call <b><i>NET_DVR_StartRemoteConfig</i></b> with "NET_DVR_JSON_CONFIG" (command No.: 2550) and set the input parameter <b>lpInBuffer</b> to the request URI: POST <b>/ISAPI/AccessControl/FingerPrintUpload?format=json</b> for setting up persistent connection, and set callback function <b>fRemoteConfigCallback</b> for getting the fingerprint information.</li><li>Call <b><i>NET_DVR_SendRemoteConfig</i></b>, set the data type <b>dwDataType</b> to "ENUM_SEND_JSON_DATA", and set the <b>pSendBuf</b> to the message <b>JSON_FingerPrintCond</b> for sending the fingerprint search condition via the persistent connection.</li></ol> <p>The fingerprint data is returned in the message <b>JSON_FingerPrintInfo</b> by the output buffer (<b>lpBuffer</b>) of the callback function <b>fRemoteConfigCallback</b> of <b><i>NET_DVR_StartRemoteConfig</i></b>.</p> <ol style="list-style-type: none"><li>Call <b><i>NET_DVR_StopRemoteConfig</i></b> to disconnect the persistent connection and finish getting.</li></ol>

	<p>Method 2:</p> <p>Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit the request URI: POST <b>/ISAPI/AccessControl/FingerPrintUpload?format=json</b> to get a fingerprint or all fingerprints of a person.</p>
<b>Delete Fingerprint</b>	<p>Method 1:</p> <ol style="list-style-type: none"><li>Call <b><u>NET_DVR_StartRemoteConfig</u></b> with "NET_DVR_DEL_FINGERPRINT_CFG_V50" (command No.: 2517) and set the input parameter <b>lpInBuffer</b> to the structure <b><u>NET_DVR_FINGER_PRINT_INFO_CTRL_V50</u></b> for setting up the persistent connection and set the callback function (<b>fRemoteConfigCallback</b>) for deleting the fingerprint information and returning the deleting status.</li><li>Check the deleting status in the structure <b><u>NET_DVR_FINGER_PRINT_STATUS_V50</u></b> returned by the output buffer (<b>lpBuffer</b>) of the callback function <b>fRemoteConfigCallback</b> of <b><u>NET_DVR_StartRemoteConfig</u></b>. When the parameter <b>byTotalStatus</b> in the structure is "1", it indicates that the fingerprint is deleted.</li><li>Call <b><u>NET_DVR_StopRemoteConfig</u></b> to disconnect the persistent connection and finish deleting.</li></ol> <p>Method 2:</p> <ol style="list-style-type: none"><li>Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit the request URI: GET <b>/ISAPI/AccessControl/FingerPrint/Delete/capabilities?format=json</b> for getting the deleting capability to know the supported deleting modes (by person or by fingerprint module) and other configuration details.</li><li>Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit the request URI: PUT <b>/ISAPI/AccessControl/FingerPrint/Delete?format=json</b> and set the input parameter <b>lpInputParam</b> to the message <b><u>JSON_FingerPrintDelete</u></b> for deleting the fingerprint information.</li><li>Call <b><u>NET_DVR_STDXMLConfig</u></b> to transmit the request URI: GET <b>/ISAPI/AccessControl/FingerPrint/DeleteProcess?format=json</b> to get the fingerprint deleting status and check whether the deleting is completed.</li></ol>

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out of the device and release the resources.

### 2.4.1 Fingerprint Collection

The fingerprint information for further management and applying should be collected by fingerprint recorder first. The following contents are about the process and parameter settings of fingerprint collection.

- Call **NET\_DVR\_GetDeviceAbility**, set the capability type **dwAbilityType** to "ACS\_ABILITY", and set the input parameter pointer **lpInBuf** to the message **XML\_Desc\_AcsAbility** for getting the access control capability to check whether fingerprint collection is supported.

- The capability is returned in the message **XML\_AcsAbility** by the output parameter pointer **pOutBuf**. The related node is <CaptureFingerPrint>.
- b. Get fingerprint collection capability to know supported parameters.
- Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CaptureFingerPrint/capabilities** to get the fingerprint collection capability.  
And the capability is returned in the message **XML\_Cap\_CaptureFingerPrint** by the output parameter (**lpOutputParam**).
  - Call **NET\_DVR\_GetDeviceAbility**, set **dwAbilityType** to "ACS\_ABILITY", and set **plnBuf** to **XML\_Desc\_AcsAbility** for getting the access control capability to know the supported fingerprint collection parameters.  
The capability is returned in the message **XML\_AcsAbility** by the output parameter pointer **pOutBuf**. The related node is <CaptureFingerPrint>.
- c. Collect the fingerprint information.
- Call **NET\_DVR\_StartRemoteConfig** with **NET\_DVR\_CAPTURE\_FINGERPRINT\_INFO** (command No.: 2504) and set **lpInBuffer** to **NET\_DVR\_CAPTURE\_FINGERPRINT\_COND** for setting up persistent connection and set callback function (**fRemoteConfigCallback**) for collecting fingerprint information.  
The collected fingerprint data is returned in the structure **NET\_DVR\_CAPTURE\_FINGERPRINT\_CFG** by **lpBuffer** of the callback function.
  - Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: POST **/ISAPI/AccessControl/CaptureFingerPrint** and set **lpInBuffer** of the input parameter (**lpInputParam**) to the message **XML\_CaptureFingerPrintCond** to collect the fingerprint information.
- d. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish collecting.

## 2.5 Manage Face Information

If a person wants to access by face, you should collect face data via the face capture module of the access control device first, create face picture libraries, and then apply face records (including face record ID, information about the person in the picture, and so on) to face picture libraries for getting the access permission.

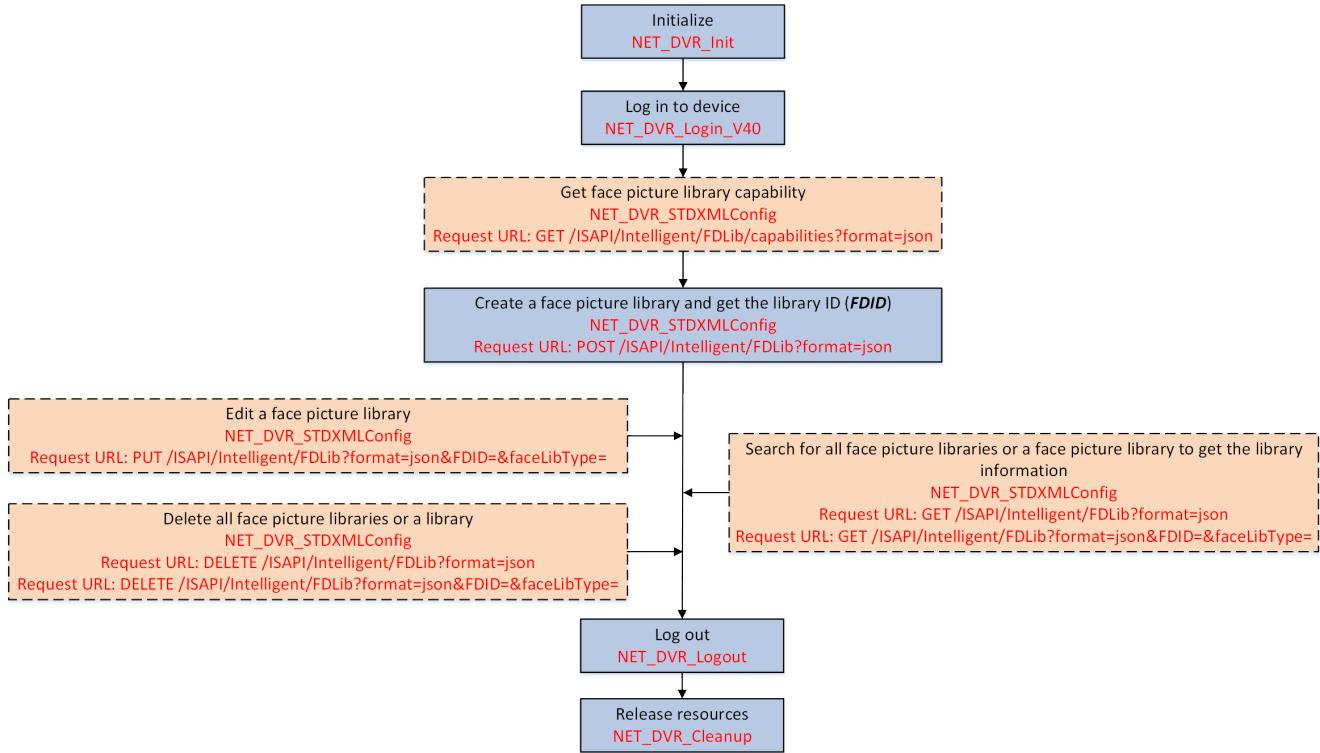
### 2.5.1 Create Face Picture Library

The face picture library refers to the library of face pictures, including captured picture library, resident population library, blocklist library, etc. You can create, edit, delete, and search for the face picture libraries.

#### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps



**Figure 1-6 Programming Flow of Creating Face Picture Library**

1. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [GET /ISAPI/Intelligent/FDLib/capabilities?format=json](#) to get the face picture library capability and check the supported operations of face picture libraries.  
The face picture library capability is returned in the message [JSON\\_FPLibCap](#) by the output parameter ([IpOutputParam](#)). If the value of the node [<supportFDFunction>](#) is "post, delete, put, get", it indicates that creating, editing, deleting, and searching for face picture libraries are supported, and you can perform the following steps to implement these functions.
2. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [POST /ISAPI/Intelligent/FDLib?format=json](#) and set the input buffer ([IpInBuffer](#)) of the input parameter ([IpInputParam](#)) to the message [JSON\\_CreateFPLibCond](#) to create a face picture library.



There are three types of face picture library, including infrared face picture library, list library, and static library. So if you want to specify a face picture library, you should provide the library type and library ID together.

The ID of the created face picture library (**FDID**) is returned.

3. **Optional:** Perform the following operation(s) after creating a face picture library.

**Edit A Face Picture Library**

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [PUT /ISAPI/Intelligent/FDLib?format=json&FDID=&faceLibType=](#) and set the

	input buffer ( <b>IpInBuffer</b> ) of the input parameter ( <b>IpInputParam</b> ) to the message <b>JSON_EditFPLibInfo</b>
<b>Delete A Face Picture Library</b>	Call <b>NET_DVR_STDXMLConfig</b> to pass through the request URL: DELETE / <b>ISAPI/Intelligent/FDLib?format=json&amp;FDID=&amp;faceLibType=</b> .
<b>Delete All Face Picture Libraries</b>	Call <b>NET_DVR_STDXMLConfig</b> to pass through the request URL: DELETE / <b>ISAPI/Intelligent/FDLib?format=json</b> .
<b>Search for A Specific Face Picture Library</b>	Call <b>NET_DVR_STDXMLConfig</b> to pass through the request URL: GET / <b>ISAPI/Intelligent/FDLib?format=json&amp;FDID=&amp;faceLibType=</b> . The information of the specified face picture library is returned in the message <b>JSON_SingleFPLibInfo</b> by the output buffer ( <b>IpOutBuffer</b> ) of the output parameter ( <b>IpOutputParam</b> ).
<b>Search for All Face Picture Libraries</b>	Call <b>NET_DVR_STDXMLConfig</b> to pass through the request URL: GET / <b>ISAPI/Intelligent/FDLib?format=json</b> . The information of all face picture libraries is returned in the message <b>JSON_FPLibListInfo</b> by the output buffer ( <b>IpOutBuffer</b> ) of the output parameter ( <b>IpOutputParam</b> ).



## Note

In the URL, both the library ID (**FDID**) and the library type (**faceLibType**) are required to specify a face picture library, e.g., **/ISAPI/Intelligent/FDLib?format=json&FDID=1223344455566788&faceLibType=blackFD**.

---

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out of the device and release the resources.

## 2.5.2 Collect Face Data

The face data for further management and applying should be collected by the face capture module of the access control device first. The following contents are about the process and parameter settings of face data collection.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

### Steps

1. Call **NET\_DVR\_GetDeviceAbility**, set **dwAbilityType** to "ACS\_ABILITY", and set **pInBuf** to **XML\_Desc\_AcsAbility** for getting the access control capability to know the supported parameters of collecting face data.

The capability is returned in the message **XML\_AcsAbility** by **pOutBuf**. The related node is **<CaptureFace>**.

2. Call [\*\*NET\\_DVR\\_StartRemoteConfig\*\*](#) with "NET\_DVR\_CAPTURE\_FACE\_INFO" (command No.: 2510) and set **IpInBuffer** to the structure [\*\*NET\\_DVR\\_CAPTURE\\_FACE\\_COND\*\*](#) for setting up persistent connection and set callback function ([\*\*fRemoteConfigCallback\*\*](#)) for collecting face data.

The collected face data is returned in the structure [\*\*NET\\_DVR\\_CAPTURE\\_FACE\\_CFG\*\*](#) by **IpBuffer** of the callback function.

3. Call [\*\*NET\\_DVR\\_StopRemoteConfig\*\*](#) to disconnect the persistent connection and finishing collecting face data.

### What to do next

Call [\*\*NET\\_DVR\\_Logout\*\*](#) and [\*\*NET\\_DVR\\_Cleanup\*\*](#) to log out of the device and release the resources.

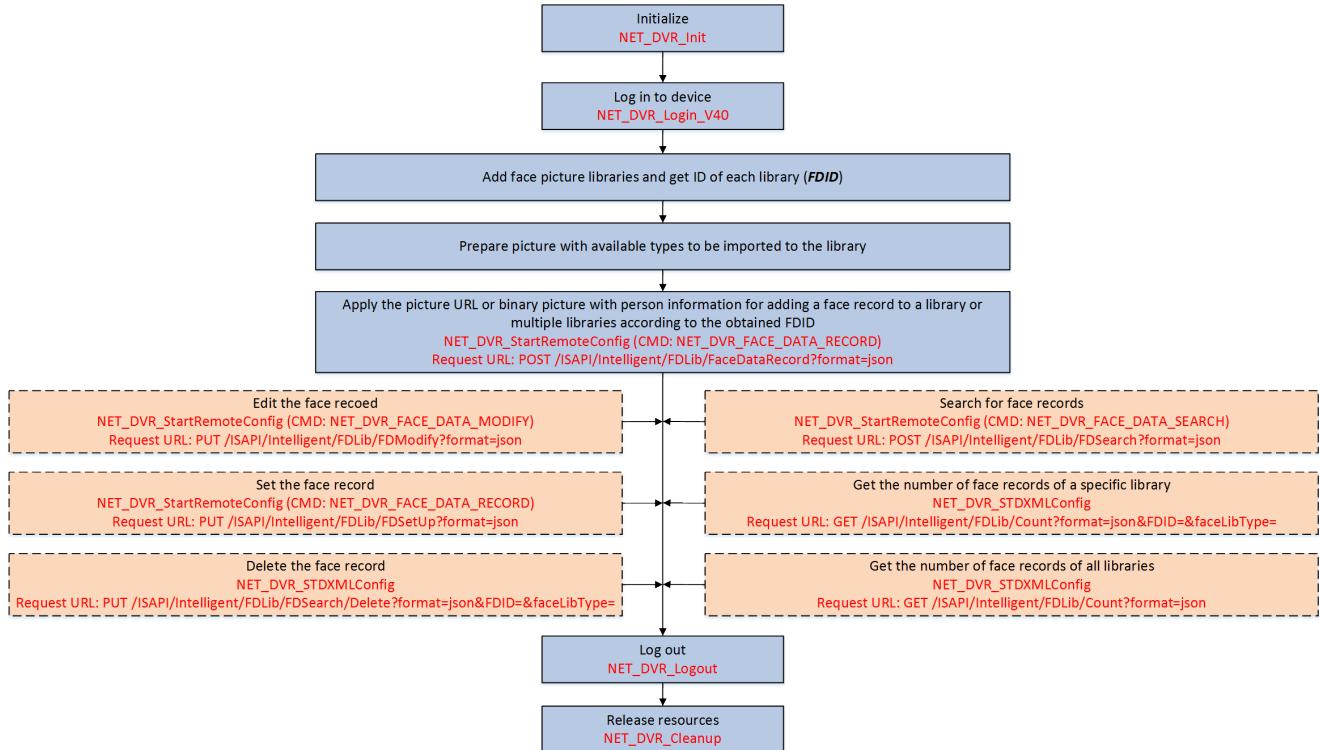
### 2.5.3 Manage Face Records in Face Picture Library

After creating face picture library, you can import face pictures with different types (i.e., picture URL and binary picture) to add the face records to the library. And you can also edit, delete, and search for the face records in the library for management.

#### Before You Start

- Make sure you have called [\*\*NET\\_DVR\\_Init\*\*](#) to initialize the development environment.
- Make sure you have called [\*\*NET\\_DVR\\_Login\\_V40\*\*](#) to log in to the device.
- Make sure you have added face picture libraries and get the ID of each library. For creating face picture library, refer to [\*\*Create Face Picture Library\*\*](#) for details.
- Make sure you have collected the face picture data, refer to [\*\*Collect Face Data\*\*](#) for details.

## Steps



**Figure 1-7 Programming Flow of Managing Face Records in Face Picture Library**

1. Prepare picture URLs (picture storage location) or binary pictures in form format for being imported to the library.
2. Apply the picture URL or binary picture with person information for adding a face record to the library according to the face picture library ID (**FDID**).
  - 1) Call **NET\_DVR\_StartRemoteConfig** with **NET\_DVR\_FACE\_DATA\_RECORD** (command No.: 2551) to pass through the request URL: **POST /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json** for setting up persistent connection and set callback function (**fRemoteConfigCallback**) for adding the face record.
  - 2) Call **NET\_DVR\_SendRemoteConfig** to send the face record information structure **NET\_DVR\_JSON\_DATA\_CFG** via the persistent connection.



### Note

- The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA".
- After a face record is added to the face picture library by calling **NET\_DVR\_SendRemoteConfig**, the next face record can be added to the face picture library only when **JSONResponseStatus** is returned by the callback function **fRemoteConfigCallback**.

- 3) Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish adding.
3. **Optional:** Perform the following operation(s) after adding face records to the face picture library.

<b>Edit Face Record</b>	<ol style="list-style-type: none"><li>a. Call <b><i>NET_DVR_StartRemoteConfig</i></b> with NET_DVR_FACE_DATA MODIFY (command No.: 2553) to pass through the request URL: PUT <b>/ISAPI/Intelligent/FDLib/FDModify?format=json</b> for setting up persistent connection and set callback function (<b>fRemoteConfigCallback</b>) for editing the face record.</li><li>b. Call <b><i>NET_DVR_SendRemoteConfig</i></b> to send the face record information structure <b><i>NET_DVR_JSON_DATA_CFG</i></b> via the persistent connection.</li></ol>
	<p> <b>Note</b></p> <p>The type of data to be sent (<b>dwDataType</b>) should be set to "ENUM_SEND_JSON_DATA".</p>
<b>Set Face Record</b>	<ol style="list-style-type: none"><li>c. Call <b><i>NET_DVR_StopRemoteConfig</i></b> to disconnect the persistent connection and finish editing.</li></ol>
	<ol style="list-style-type: none"><li>a. Call <b><i>NET_DVR_StartRemoteConfig</i></b> with NET_DVR_FACE_DATA_RECORD (command No.: 2551) to pass through the request URL: PUT <b>/ISAPI/Intelligent/FDLib/FDSetUp?format=json</b> for setting up persistent connection and set callback function (<b>fRemoteConfigCallback</b>) for setting the face record.</li><li>b. Call <b><i>NET_DVR_SendRemoteConfig</i></b> to send the face record information structure <b><i>NET_DVR_JSON_DATA_CFG</i></b> via the persistent connection.</li></ol>
	<p> <b>Note</b></p> <p>The type of data to be sent (<b>dwDataType</b>) should be set to "ENUM_SEND_JSON_DATA".</p>
<b>Delete Face Record(s)</b>	<ol style="list-style-type: none"><li>c. Call <b><i>NET_DVR_StopRemoteConfig</i></b> to disconnect the persistent connection and finish setting.</li></ol>
	<p>Call <b><i>NET_DVR_STDXMLConfig</i></b> to pass through the request URL: PUT <b>/ISAPI/Intelligent/FDLib/FDSearch/Delete?format=json&amp;FDID=&amp;faceLibType=</b> and set the input buffer (<b>lpInBuffer</b>) of the input parameter (<b>lpInputParam</b>) to the message <b><i>JSON_DelFaceRecord</i></b>.</p>
	<p> <b>Note</b></p> <p>Deleting a face record or deleting face records in a batch are both supported.</p>
<b>Search for Face Records</b>	<ol style="list-style-type: none"><li>a. Call <b><i>NET_DVR_StartRemoteConfig</i></b> with NET_DVR_FACE_DATA_SEARCH (command No.: 2552) to pass through the request URL: POST <b>/ISAPI/Intelligent/FDLib/FDSearch?format=json</b> for setting up persistent connection and set callback function (<b>fRemoteConfigCallback</b>) for searching for face records.</li><li>b. Call <b><i>NET_DVR_SendRemoteConfig</i></b> to send the search condition message <b><i>JSON_SearchFaceRecordCond</i></b> via the persistent connection.</li></ol>



## Note

The type of data to be sent (**dwDataType**) should be set to "ENUM\_SEND\_JSON\_DATA", and the search result message **JSON\_SearchFaceRecordResult** will be returned in the callback function configured by the above steps.

- c. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish searching.
- 



## Note

Searching multiple face picture libraries at a time and fuzzy search are both supported.

---

### Get Number of Face

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET [\*\*/ISAPI/Intelligent/FDLib/Count?format=json&FDID=&faceLibType=\*\*](#).

### Records of A Specific Face Picture Library

The result is returned in the message **JSON\_FaceRecordNumInOneFPLib** by the output buffer (**lpOutBuffer**) of the output parameter (**lpOutputParam**).

### Get Number of Face Records of

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET [\*\*/ISAPI/Intelligent/FDLib/Count?format=json\*\*](#).

### All Face Picture Libraries

The result is returned in the message **JSON\_FaceRecordNumInAllFPLib** by the output buffer (**lpOutBuffer**) of the output parameter (**lpOutputParam**).

---



## Note

In the request URL, both the library ID (**FDID**) and library type (**faceLibType**) are required to specify a face picture library, e.g., [\*\*/ISAPI/Intelligent/FDLib?format=json&FDID=1223344455566788&faceLibType=blackFD\*\*](#).

---

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out of the device and release the resources.

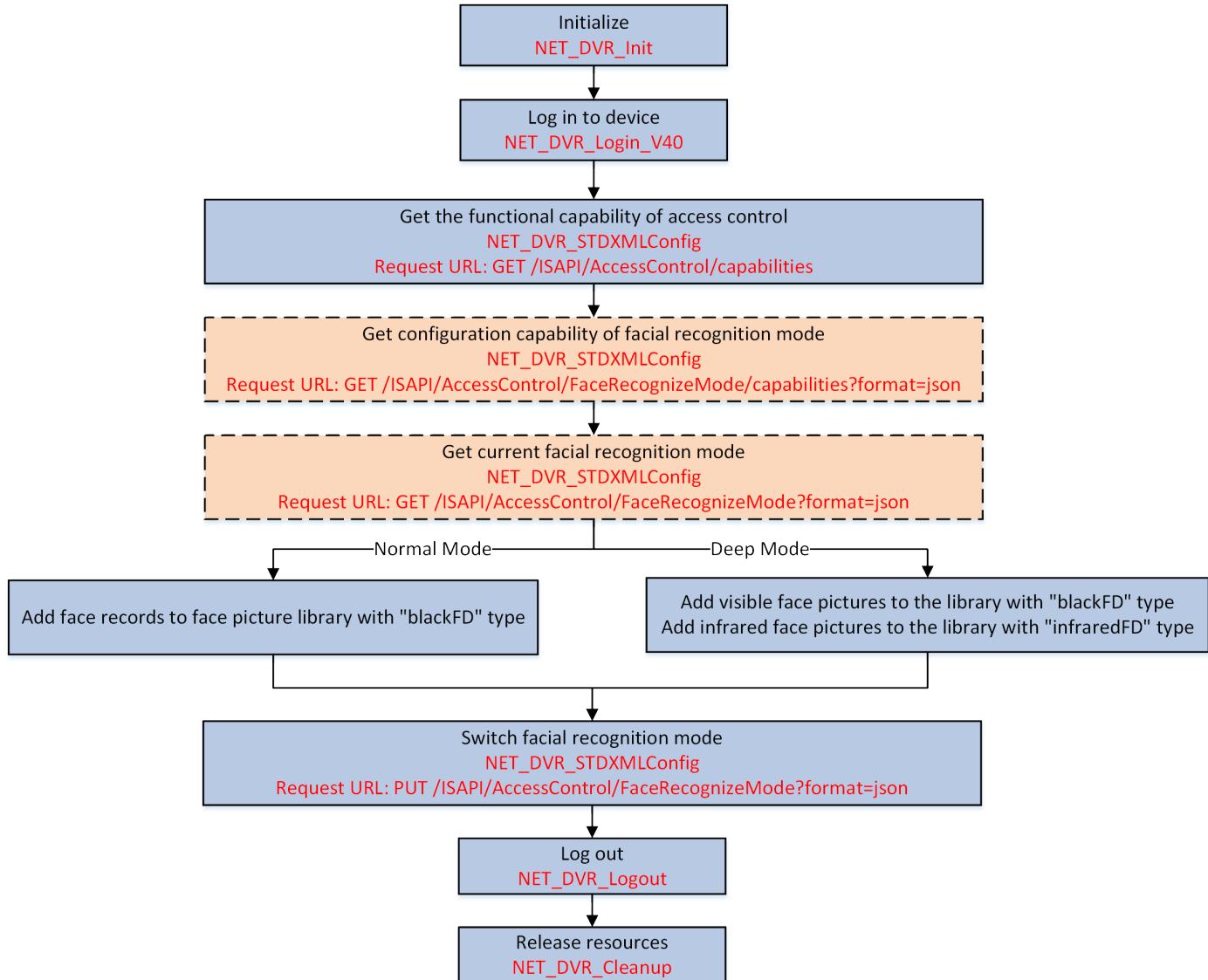
## 2.5.4 Configure Facial Recognition Mode

When recognizing human faces via the access control device, both the normal mode and the deep mode are available. For the normal mode, the human face is recognized via white light camera; for the deep mode, the human face is recognized by the IR light camera, which is applicable to a more complicated environment and can recognize a much wider people range than the normal mode.

### Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to the device.

## Steps



**Figure 1-8 Programming Flow of Configuring Facial Recognition Mode**

- Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: **GET /ISAPI/AccessControl/capabilities** to get the functional capability of access control and check whether the device supports configuring facial recognition mode.



The capability will be returned in the message **XML\_Cap\_AccessControl** by **IpOutputParam**. If the device supports configuring facial recognition mode, the node **<isSupportFaceRecognizeMode>** is returned and its value is "true", and then you can perform the following steps. Otherwise, it indicates that configuring facial recognition mode is not supported by the device, please end this task.

2. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/FaceRecognizeMode/capabilities?format=json](#) to get the configuration capability of facial recognition mode to know the supported facial recognition modes.  
The configuration capability is returned in the message [JSON\\_Cap\\_FaceRecognizeMode](#) by [IpOutputParam](#).
3. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/FaceRecognizeMode?format=json](#) to get the current facial recognition mode.  
The parameters of the current facial recognition mode are returned in the message [JSON\\_FaceRecognizeMode](#) by [IpOutBuffer](#) of [IpOutputParam](#).
4. Perform one of the following operations to add face records to face picture libraries for setting normal or deep facial recognition mode.
- Add face records to the default face picture library with "blackFD" type, refer to [Manage Face Records in Face Picture Library](#) for details.
  - Add visible face pictures to the default face picture library with "blackFD" type, and add infrared face pictures to the default library with "infraredFD" type, refer to [Manage Face Records in Face Picture Library](#) for details.



### Note

Generally, during the initialization of the access control device, two face picture libraries with "blackID" type (the library ID is 1) and "infraredFD" type (the library ID is 2) will be created automatically. But if the default libraries have not been created, you should create them by yourself, refer to [Create Face Picture Library](#) for details.

5. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/FaceRecognizeMode?format=json](#) and set [IpInBuffer](#) of [IpInputParam](#) to the message [JSON\\_FaceRecognizeMode](#) to configure facial recognition mode.

### Result

The device will reboot automatically after configuring facial recognition mode, and permissions linked with face pictures in the library will be cleared.

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out of the device and release the resources.

## 2.5.5 Other Facial Applications

### Verify Face Pictures in a Batch

1. Call [NET\\_DVR\\_GetDeviceAbility](#), set the capability type [dwAbilityType](#) to "ACS\_ABILITY" (macro definition value: 0x801), and set the input pointer ([pInBuf](#)) to [XML\\_Desc\\_AcsAbility](#) for getting the access control capability to check whether verifying face pictures in a batch is supported.  
The capability is returned in the message [XML\\_AcsAbility](#) by the output pointer ([pOutBuf](#)).



## Note

If the node <CheckFacePicture> is returned, it indicates that verifying face pictures in a batch is supported, then you can continue to perform the following steps.

2. Call **NET\_DVR\_StartRemoteConfig** with "NET\_DVR\_BULK\_CHECK\_FACE\_PICTURE" (command No.: 2533) and set the input parameter **lpInBuffer** to the structure **NET\_DVR\_CHECK\_FACE\_PICTURE\_COND** for setting up the persistent connection and set the callback function **fRemoteConfigCallback** for getting the verification result.
  3. Call **NET\_DVR\_SendRemoteConfig**, set the data type (**dwDataType**) to "ENUM\_ACS\_SEND\_DATA" (macro definition value: 0x3), and set the sending buffer (**pSendBuf**) to **NET\_DVR\_CHECK\_FACE\_PICTURE\_CFG** for sending the face picture to be verified via the persistent connection.
- 



## Note

The verification result is returned in the structure **NET\_DVR\_CHECK\_FACE\_PICTURE\_STATUS** by the output buffer (**lpBuffer**) of the callback function.

4. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish verifying face pictures in a batch.

## Get Face Modeling Failure Information after Upgrading Device

1. Call **NET\_DVR\_GetDeviceAbility**, set the capability type **dwAbilityType** to "ACS\_ABILITY", and set the input buffer (**lpInBuf**) to **XML\_Desc\_AcsAbility** for getting the access control capability to check if getting the information of face modeling failure after upgrading the device is supported. The capability is returned in the message **XML\_AcsAbility** by the output pointer (**pOutBuf**).
- 



## Note

If the node <isSupportGetFailedFaceInfo> is returned and is set to "true", it indicates that getting the information of face modeling failure after upgrading the device is supported, and you can continue to perform the following steps.

2. Call **NET\_DVR\_StartRemoteConfig** with **NET\_DVR\_GET\_FAILED\_FACE\_INFO** (command No.: 2522) and set the input parameter **lpInBuffer** to the structure **NET\_DVR\_FAILED\_FACE\_COND** to set up persistent connection and set callback function **fRemoteConfigCallback** for getting the information.
3. Call **NET\_DVR\_StopRemoteConfig** to disconnect the persistent connection and finish getting the information.

## Configure Conditions for Face Picture Comparison

- Get Condition Configuration Capability of Face Picture Comparison  
Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET [/ISAPI/AccessControl/FaceCompareCond/capabilities](https://ISAPI/AccessControl/FaceCompareCond/capabilities).

And the configuration capability is returned in the message [XML\\_Cap\\_FaceCompareCond](#) by IpOutputParam.

- Get Conditions of Face Picture Comparison

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/FaceCompareCond](#).

And the conditions are returned in the message [XML\\_FaceCompareCond](#) by IpOutputParam.

- Set Conditions for Face Picture Comparison

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/FaceCompareCond](#) and set IpInputParam to [XML\\_FaceCompareCond](#).

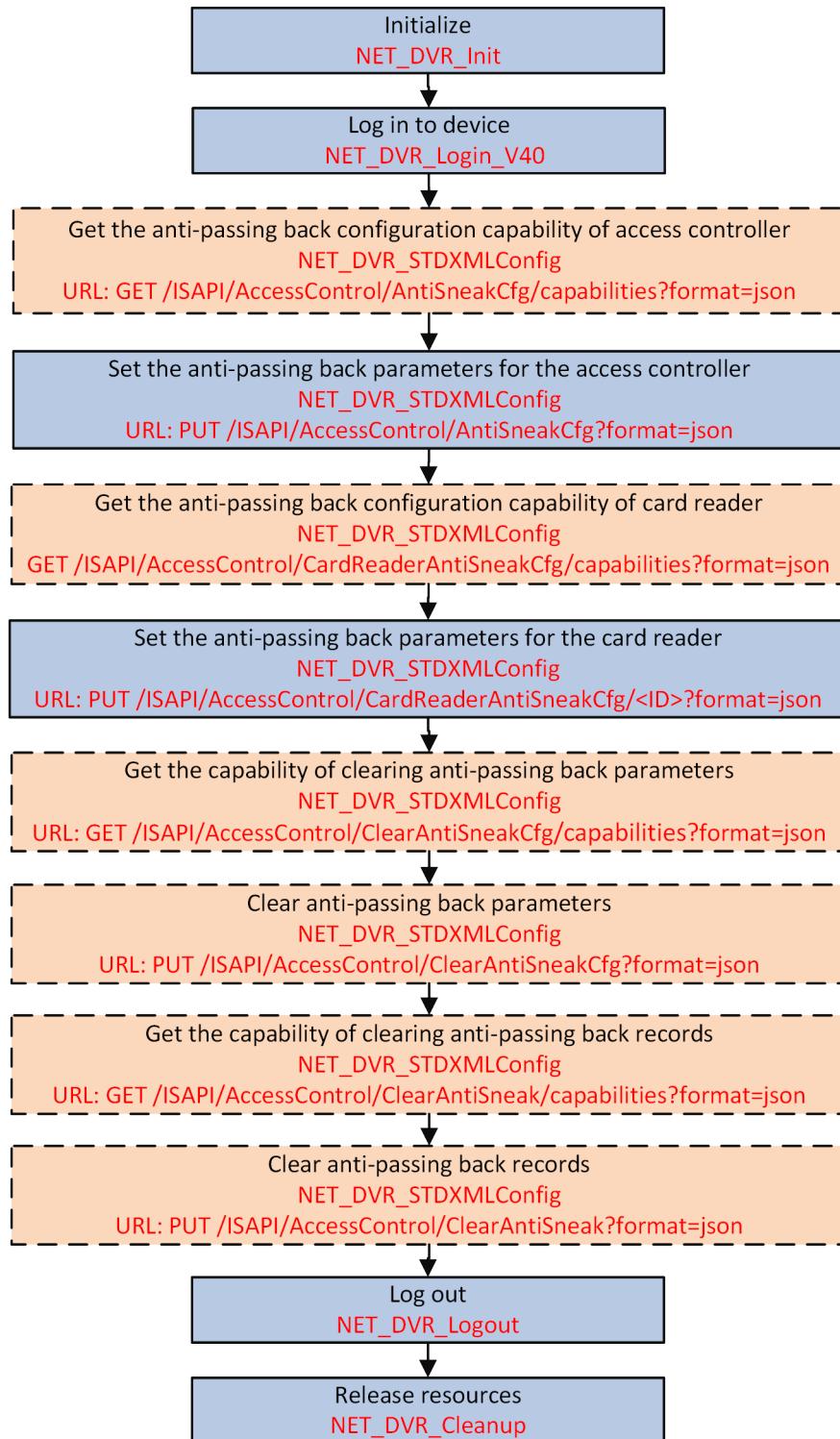
## 2.6 Configure Anti-Passing Back

The anti-passing back is to set the only route for passing the access control points and only one person could pass after swiping card. You can configure this function to enhance the access security of some important and specific places (e.g., laboratories, offices).

### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to the device.

## Steps



**Figure 1-9 Programming Flow of Configuring Anti-Passing Back**



Before setting the following parameters, you'd better pass through the each configuration URLs with GET method to get the existing or configured parameters for reference.

1. **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/AntiSneakCfg/capabilities?format=json** for getting the anti-passing back configuration capability of the access controller.  
The anti-passing back configuration capability **JSON\_Cap\_AntiSneakCfg** is returned.
2. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/AntiSneakCfg?format=json** for setting the anti-passing back parameters of access controller.
3. **Optional:** Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/CardReaderAntiSneakCfg/capabilities?format=json** for getting the anti-passing back configuration capability of card reader.
4. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/CardReaderAntiSneakCfg/<ID>?format=json** for setting the anti-passing parameters of card reader.
5. Perform the following operation(s) after configuring the anti-passing back function.

**Clear Anti-passing Back Parameters**

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/ClearAntiSneakCfg?format=json**.



The capability of clearing anti-passing back parameters (**JSON\_Cap\_ClearAntiSneak**) can be obtained by calling **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/ClearAntiSneakCfg/capabilities?format=json**.

**Clear Anti-passing Back Records**

If the anti-passing back event occurred, it will be recorded in the access controller, so if needed, you can call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/ClearAntiSneak?format=json** for clearing the records.



The capability of clearing anti-passing back records (**JSON\_Cap\_ClearAntiSneak**) can be obtained by calling **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/ClearAntiSneak/capabilities?format=json**.

## What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release the resources.

## 2.7 Cross-Controller Anti-Passing Back Configuration

You can set anti-passing back for card readers in multiple access controllers. You should swipe the card according to the configured swiping card route or entrance/exit. And only one person could pass the access control point after swiping the card.

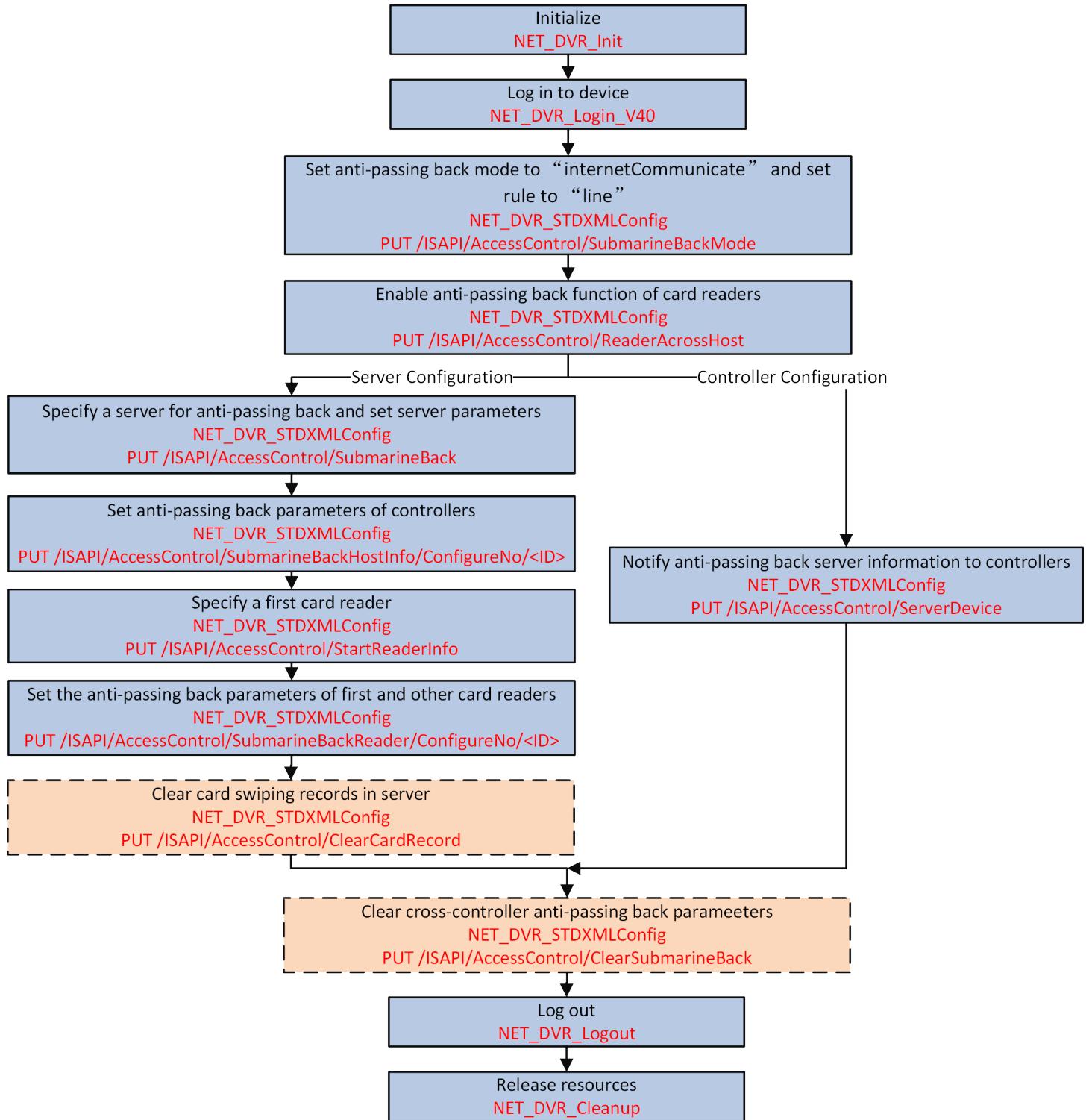
### 2.7.1 Configure Route Anti-Passing Back Based on Network

The route anti-passing back depends on the card swiping route. You should set the first card reader and the card readers afterwards. It will authenticate the anti-passing back according to the entrance and exit information stored on the card reader.

#### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to the device.

## Steps



**Figure 1-10 Programming Flow of Configuring Route Anti-Passing Back Based on Network**



Before setting the following parameters, you'd better pass through the each configuration URLs with GET method to get the existing or configured parameters for reference.

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/SubmarineBackMode** for setting anti-passing back mode and rule.



- For route anti-passing back based on network, the mode must be set to "internetCommunicate" and the rule should be set to "line".
- To get the capability of setting anti-passing back mode and rule, you should pass through the request URL: GET **/ISAPI/AccessControl/SubmarineBackMode/capabilities**. And the capability is returned in the message **XML\_Cap\_SubmarineBackMode**.

2. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/ReaderAcrossHost** for enabling anti-passing back of card readers.



To get the capability of enabling anti-passing back of card readers, you should pass through the request URL: GET **/ISAPI/AccessControl/ReaderAcrossHost/capabilities**. And the capability is returned in the message **XML\_Cap\_ReaderAcrossHost**.

3. Perform one of the following operations to configure anti-passing back server or access controllers.

- Configure anti-passing back server:

- a. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/SubmarineBack** for specifying an access controller as the server for cross-controller anti-passing back and setting the server parameters.



To get the capability of specifying a server for anti-passing back, you should pass through the request URL: GET **/ISAPI/AccessControl/SubmarineBack/capabilities**. And the capability is returned in the message **XML\_Cap\_SubmarineBack**.

- b. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID>** for setting anti-passing back parameters of access controllers.



To get the capability of adding access controllers to anti-passing back route, you should pass through the request URL: GET **/ISAPI/AccessControl/SubmarineBackHostInfo/capabilities**. And the capability is returned in the message **XML\_Cap\_SubmarineBackHostInfo**.

- c. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/StartReaderInfo** for specifying a first card reader.



### Note

To get the capability of specifying a first card reader, you should pass through the request URL: GET [/ISAPI/AccessControl/StartReaderInfo/capabilities](#). And the capability is returned in the message [XML Cap StartReaderInfo](#).

- d. Call [NET DVR STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/SubmarineBackReader/ConfigureNo/<ID>](#) for setting the anti-passing back parameters of the first and other card readers.
- 



### Note

To get the capability of setting anti-passing back parameters for card readers, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackReader/capabilities](#). And the capability is returned in the message [XML Cap SubmarineBackReader](#).

- Call [NET DVR STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ServerDevice](#) for notifying the anti-passing back server information to access controllers.
- 



### Note

To get the capability of notifying anti-passing back server information to access controllers, you should pass through the request URL: GET [/ISAPI/AccessControl/ServerDevice/capabilities](#). And the capability is returned in the message [XML Cap ServerDevice](#).

4. **Optional:** Perform the following operation(s) after configuring route anti-passing back based on network.

**Clear Cross-Controller Anti-Passing Back Parameters**

Call [NET DVR STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ClearSubmarineBack](#).

---



### Note

To get the capability of clearing the cross-controller anti-passing back parameters, you should pass through the request URL: GET [/ISAPI/AccessControl/ClearSubmarineBack/capabilities](#). And the capability is returned in message [XML Cap ClearSubmarineBack](#)

---

**Clear Card Swiping Records in Server**

If the card is swiped in the anti-passing back route or entrance/exit, it will be recorded by the server. So if need, you can call [NET DVR STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ClearCardRecord](#) for clearing card swiping records in the server.

---



### Note

To get the capability of clearing card swiping records in server, you should pass through the request URL: GET [/ISAPI/AccessControl/](#)

*ClearCardRecord/capabilities* . And the capability is returned in message *XML\_Cap\_ClearCardRecord*

---

### What to do next

Call *NET\_DVR\_Logout* and *NET\_DVR\_Cleanup* to log out and release the resources.

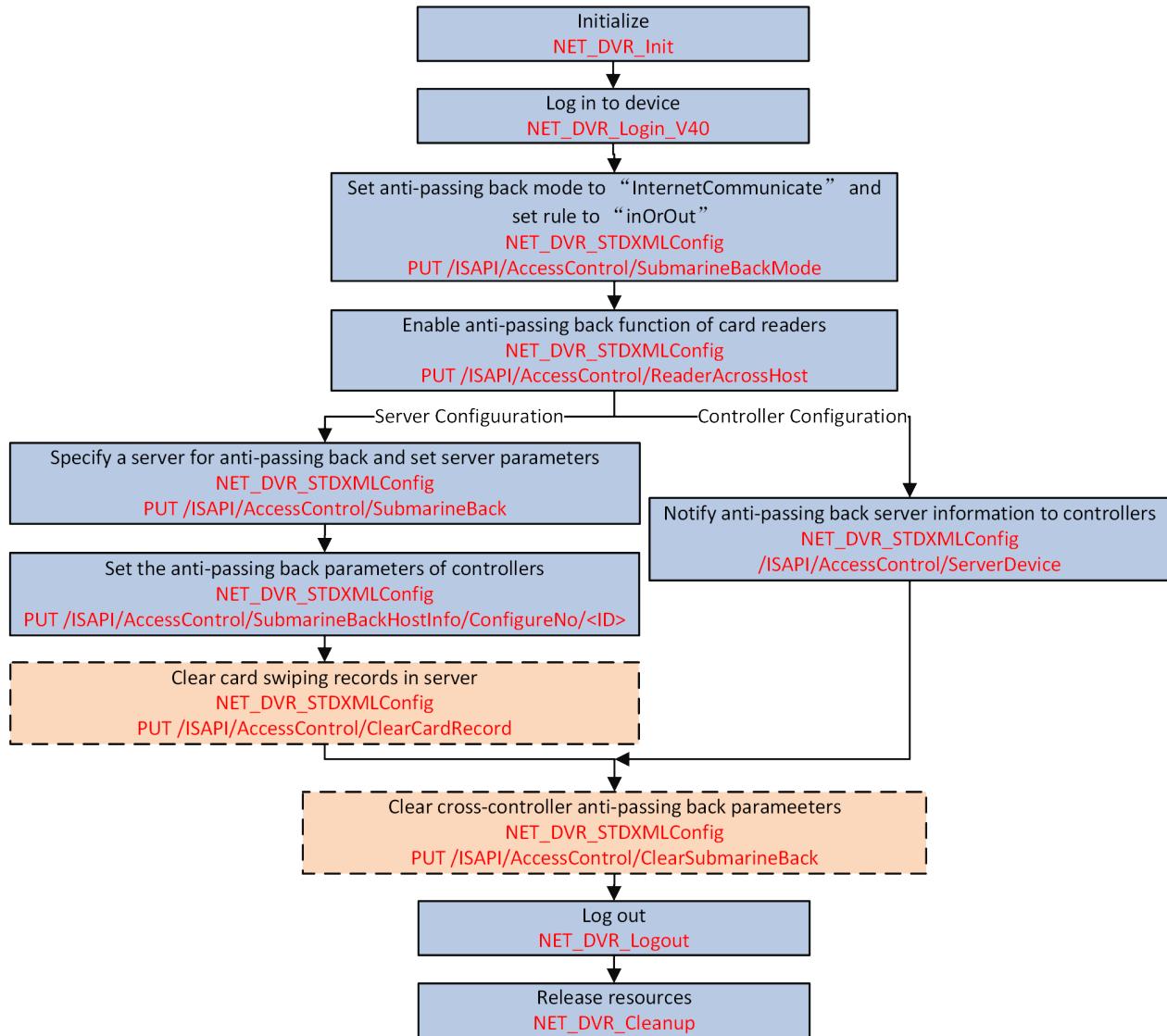
### 2.7.2 Configure Entrance/Exit Anti-Passing Back Based on Network

You can set the entrance card reader and the exit card reader only for entering and exiting, without setting the first card reader and the card readers afterwards. It will authenticate the anti-passing back according to the entrance and exit information on the card reader.

#### Before You Start

- Make sure you have called *NET\_DVR\_Init* to initialize the development environment.
- Make sure you have called *NET\_DVR\_Login\_V40* to log in to the device.

## Steps



**Figure 1-11 Programming Flow of Configuring Entrance/Exit Anti-Passing Back Based on Network**



Before setting the following parameters, you'd better pass through the each configuration URLs with GET method to get the existing or configured parameters for reference.

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: **PUT /ISAPI/AccessControl/SubmarineBackMode** for setting anti-passing back mode and rule.



## Note

- For route anti-passing back based on network, the mode must be set to "internetCommunicate" and the rule should be set to "inOrOut".
- To get the capability of setting anti-passing back mode and rule, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackMode/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBackMode](#).

2. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ReaderAcrossHost](#) for enabling anti-passing back of card readers.

---



## Note

To get the capability of enabling anti-passing back of card readers, you should pass through the request URL: GET [/ISAPI/AccessControl/ReaderAcrossHost/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_ReaderAcrossHost](#).

---

3. Perform one of the following operations to configure anti-passing back server or access controllers.

- Configure anti-passing back server:

a. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/SubmarineBack](#) for specifying an access controller as the server for cross-controller anti-passing back and setting the server parameters.

---



## Note

To get the capability of specifying a server for anti-passing back, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBack/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBack](#).

b. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID>](#) for setting anti-passing back parameters of access controllers.

---



## Note

To get the capability of adding access controllers to anti-passing back route, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackHostInfo/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBackHostInfo](#).

- Configure the access controllers:

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ServerDevice](#) for notifying the anti-passing back server information to access controllers.

---



## Note

To get the capability of notifying anti-passing back server information to access controllers, you should pass through the request URL: GET [/ISAPI/AccessControl/ServerDevice/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_ServerDevice](#).

---

**4. Optional:** Perform the following operation(s) after configuring entrance/exit anti-passing back based on network.

**Clear Cross-Controller Anti-Passing Back Parameters**

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ClearSubmarineBack](#).

---



**Note**

To get the capability of clearing the cross-controller anti-passing back parameters, you should pass through the request URL: GET [/ISAPI/AccessControl/ClearSubmarineBack/capabilities](#). And the capability is returned in message [XML\\_Cap\\_ClearSubmarineBack](#)

---

**Clear Card Swiping Records in Server**

If the card is swiped in the anti-passing back route or entrance/exit, it will be recorded by the server. So if need, you can call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ClearCardRecord](#) for clearing card swiping records in the server.

---



**Note**

To get the capability of clearing card swiping records in server, you should pass through the request URL: GET [/ISAPI/AccessControl/ClearCardRecord/capabilities](#). And the capability is returned in message [XML\\_Cap\\_ClearCardRecord](#)

---

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release the resources.

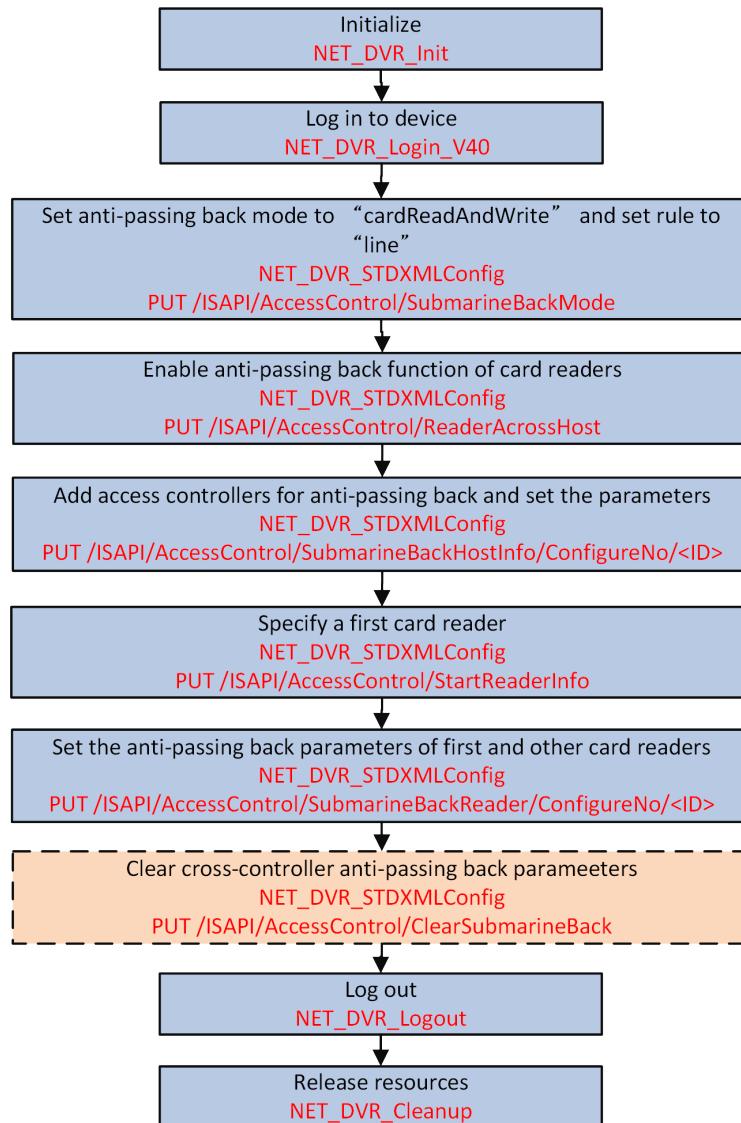
### 2.7.3 Configure Route Anti-Passing Back Based on Card

The route anti-passing back depends on the card swiping route. You should set the first card reader and the card readers afterwards. It will judge the anti-passing back according to the entrance and exit records on the card.

#### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to the device.

## Steps



**Figure 1-12 Programming Flow of Configuring Route Anti-Passing Back Based on Card**



Before setting the parameters, you'd better pass through the each configuration URLs with GET method to get the existing or configured parameters for reference.

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: **PUT /ISAPI/AccessControl/SubmarineBackMode** for setting anti-passing back mode and rule.



## Note

- For route anti-passing back based on network, the mode must be set to "cardReadAndWrite" and the rule should be set to "line".
- To get the capability of setting anti-passing back mode and rule, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackMode/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBackMode](#).

2. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ReaderAcrossHost](#) for enabling anti-passing back of card readers.

---



## Note

To get the capability of enabling anti-passing back of card readers, you should pass through the request URL: GET [/ISAPI/AccessControl/ReaderAcrossHost/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_ReaderAcrossHost](#).

3. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID>](#) for adding access controllers for anti-passing back and set their parameters.

---



## Note

To get the capability of adding access controllers to anti-passing back route, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackHostInfo/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBackHostInfo](#).

4. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/StartReaderInfo](#) for specifying a first card reader.

---



## Note

To get the capability of specifying a first card reader, you should pass through the request URL: GET [/ISAPI/AccessControl/StartReaderInfo/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_StartReaderInfo](#).

5. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/SubmarineBackReader/ConfigureNo/<ID>](#) for setting the anti-passing back parameters of the first and other card readers.

---



## Note

To get the capability of setting anti-passing back parameters for card readers, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackReader/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBackReader](#).

6. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ClearSubmarineBack](#) for clearing the cross-controller anti-passing back parameters.

---



## Note

To get the capability of clearing the cross-controller anti-passing back parameters, you should pass through the request URL: GET [/ISAPI/AccessControl/ClearSubmarineBack/capabilities](#) . And the capability is returned in message [XML\\_Cap\\_ClearSubmarineBack](#)

---

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release the resources.

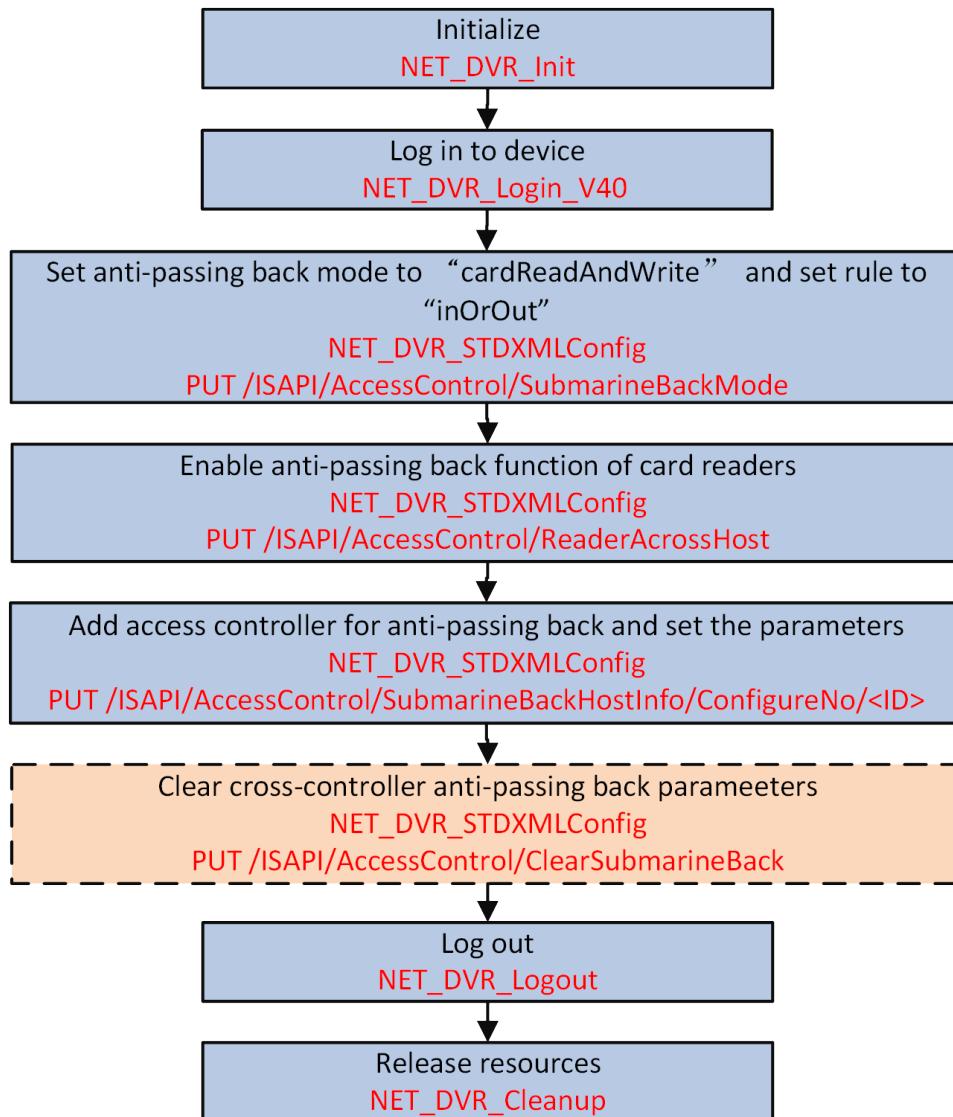
## 2.7.4 Configure Entrance/Exit Anti-Passing Back Based on Card

You can set the entrance card reader and the exit card reader only for entering and exiting, without setting the first card reader and the card readers afterward. It will judge the anti-passing back according to the entrance and exit records on the card.

### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to the device.

## Steps



**Figure 1-13 Programming Flow of Configuring Entrance/Exit Anti-Passing Back Based on Card**



Before setting the following parameters, you'd better pass through the each configuration URLs with GET method to get the existing or configured parameters for reference.

1. Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: `PUT /ISAPI/AccessControl/SubmarineBackMode` for setting anti-passing back mode and rule.



## Note

- For route anti-passing back based on network, the mode must be set to "cardReadAndWrite" and the rule should be set to "inOrOut".
- To get the capability of setting anti-passing back mode and rule, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackMode/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBackMode](#).

2. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ReaderAcrossHost](#) for enabling anti-passing back of card readers.

---



## Note

To get the capability of enabling anti-passing back of card readers, you should pass through the request URL: GET [/ISAPI/AccessControl/ReaderAcrossHost/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_ReaderAcrossHost](#).

3. Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID>](#) for adding access controllers for anti-passing back entrance/exit and set their parameters.

---



## Note

To get the capability of adding access controllers to anti-passing back entrance/exit, you should pass through the request URL: GET [/ISAPI/AccessControl/SubmarineBackHostInfo/capabilities](#). And the capability is returned in the message [XML\\_Cap\\_SubmarineBackHostInfo](#).

4. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/ClearSubmarineBack](#) for clearing the cross-controller anti-passing back parameters.

---



## Note

To get the capability of clearing the cross-controller anti-passing back parameters, you should pass through the request URL: GET [/ISAPI/AccessControl/ClearSubmarineBack/capabilities](#). And the capability is returned in message [XML\\_Cap\\_ClearSubmarineBack](#).

## What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release the resources.

## 2.8 Schedule Settings

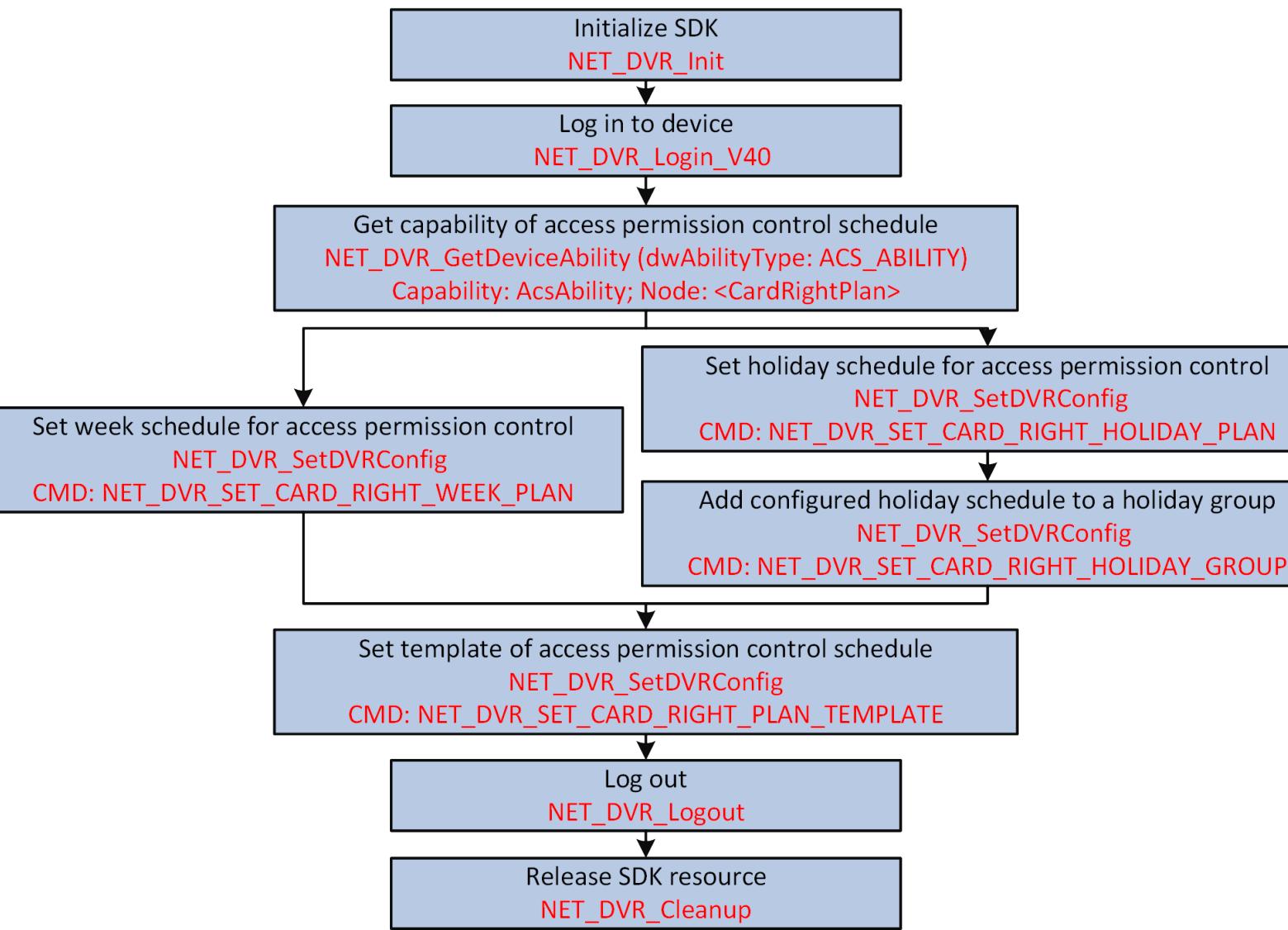
### 2.8.1 Configure Access Permission Control Schedule

To regularly control the access permissions for managing the accessible time duration (by default, it is 24 hours) of some important access control points, you can configure the week or holiday schedules.

## Before You Start

- Make sure you have called **NET\_DVR\_Init** to initialize the development environment.
- Make sure you have called **NET\_DVR\_Login\_V40** to log in to device.

## Steps



**Figure 1-14 Programming Flow of Configuring Access Permission Control Schedule**

1. Call **NET\_DVR\_GetDeviceAbility**, specify **dwAbilityType** to "ACS\_ABILITY", set **pInBuf** to **XML\_Desc\_AcsAbility** for getting the access control capability to check if configuring access permission control schedule is supported.

The capability is returned in the message **XML\_AcsAbility** by **pOutBuf**.

If the node **<CardRightPlan>** is returned, it indicates that configuring access permission control schedule is supported, and you can continue to perform the following steps.

Otherwise, configuring access permission control schedule is not supported, please end this task.

2. Perform one of the following operations to set week or holiday schedule for access permission control.

- a. Call **NET\_DVR\_GetDVRConfig** with "NET\_DVR\_GET\_CARD\_RIGHT\_WEEK\_PLAN" (command No.: 2126) to get default or configured week schedule configurations for reference.



The week schedule parameters are returned in the structure **NET\_DVR\_WEEK\_PLAN\_CFG** by **IpOutBuffer**.

- b. Call **NET\_DVR\_SetDVRConfig** with "NET\_DVR\_SET\_CARD\_RIGHT\_WEEK\_PLAN" (command No.: 2127) and set **IpInBuffer** to **NET\_DVR\_WEEK\_PLAN\_CFG** for setting the week schedule.
- a. Call **NET\_DVR\_GetDVRConfig** with "NET\_DVR\_GET\_CARD\_RIGHT\_HOLIDAY\_PLAN" (command No.: 2130) to get default or configured holiday schedule configurations for reference.



The holiday schedule parameters are returned in the structure **NET\_DVR\_HOLIDAY\_PLAN\_CFG** by **IpOutBuffer**.

- b. Call **NET\_DVR\_SetDVRConfig** with "NET\_DVR\_SET\_CARD\_RIGHT\_HOLIDAY\_PLAN" (command No.: 2131) and set **IpInBuffer** to **NET\_DVR\_HOLIDAY\_PLAN\_CFG** for setting the week schedule.
- c. Call **NET\_DVR\_GetDVRConfig** with "NET\_DVR\_GET\_CARD\_RIGHT\_HOLIDAY\_GROUP" (command No.: 2134) to get default or configured holiday group configurations for reference.



The holiday group parameters are returned in the structure **NET\_DVR\_HOLIDAY\_GROUP\_CFG** by **IpOutBuffer**.

- d. Call **NET\_DVR\_SetDVRConfig** with "NET\_DVR\_SET\_CARD\_RIGHT\_HOLIDAY\_GROUP" (command No.: 2135) and set **IpInBuffer** to **NET\_DVR\_HOLIDAY\_GROUP\_CFG** for adding the configured holiday schedule to a holiday group.

3. **Optional:** Call **NET\_DVR\_GetDVRConfig** with "NET\_DVR\_GET\_CARD\_RIGHT\_PLAN\_TEMPLATE" (command No.: 2138) to get default or configured schedule template configurations for reference.



The schedule template parameters are returned in the structure **NET\_DVR\_PLAN\_TEMPLATE** by **IpOutBuffer**.

4. Call **NET\_DVR\_SetDVRConfig** with "NET\_DVR\_SET\_CARD\_RIGHT\_PLAN\_TEMPLATE" (command No.: 2139) and set **IpInBuffer** to **NET\_DVR\_PLAN\_TEMPLATE** for setting the schedule template.



## Note

The configured schedule template can be directly linked to person ID when applying person information. And the linked person can get the access permission configured in the template.

---

### Example

#### Sample Code for Configuring Access Permission Control Schedule

```
#include <stdio.h>
#include <iostream>
#include <afx.h>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

void main()
{
    //-----
    //Initialize
    NET_DVR_Init();

    //Set connection timeout and reconnection function
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    //Log in to device
    LONG lUserID;
    //Login parameters, including device IP address, user name, password, and
so on
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsynLogin = 0; //Synchronous login mode
    strcpy(struLoginInfo.sDeviceAddress, "192.168.1.64"); //Device IP address
    struLoginInfo.wPort = 8000; //Device service port number
    strcpy(struLoginInfo.sUserName, "admin"); //User name
    strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

    //Device information, output parameter
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

    lUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
    if (lUserID < 0)
    {
        printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }
    //-----
    //Set access permission schedule template, when issuing card, link to this
template

    CString     m_csTemplateName = "Access permission schedule template 1";
```

```

NET_DVR_PLAN_TEMPLATE struPlanTem = {0};
struPlanTem.dwSize = sizeof(struPlanTem);
struPlanTem.byEnable = 1;//Enable or not: 0-No, 1-Yes
strncpy((char *)struPlanTem.byTemplateName, (LPCTSTR)m_csTemplateName,
TEMPLATE_NAME_LEN);
struPlanTem.dwWeekPlanNo = 1;//Week schedule No.1
struPlanTem.dwHolidayGroupNo[0] = 1;//Holiday group No.1, up to 16 holiday
groups can be linked to each schedule
//struPlanTem.dwHolidayGroupNo[1] = 2;//Holiday group No.2

BOOL bRet1 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_CARD_RIGHT_PLAN_TEMPLATE, 1, \
&struPlanTem, sizeof(struPlanTem));
if (!bRet1)
{
    printf("Setting access permission schedule template failed, error:%d.
\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set week schedule 1 for access permission
NET_DVR_WEEK_PLAN_CFG struWeekPlan = {0};
struWeekPlan.dwSize = sizeof(struWeekPlan);
struWeekPlan.byEnable = 1;//Enable week schedule

NET_DVR_SINGLE_PLAN_SEGMENT struSinglePlanSegment = {0};
LPNET_DVR_SINGLE_PLAN_SEGMENT lpPlanSegment = &struSinglePlanSegment;
struSinglePlanSegment.byEnable = 1;

struSinglePlanSegment.struTimeSegment.struBeginTime.byHour = 0;//Start time
struSinglePlanSegment.struTimeSegment.struBeginTime.byMinute = 0;
struSinglePlanSegment.struTimeSegment.struBeginTime.bySecond = 0;

struSinglePlanSegment.struTimeSegment.struEndTime.byHour = 23;//End time
struSinglePlanSegment.struTimeSegment.struEndTime.byMinute = 59;
struSinglePlanSegment.struTimeSegment.struEndTime.bySecond = 59;

/*Up to 8 time periods can be set for each day. Here only takes setting one
period for each day*/

for (int iDate = 0; iDate<MAX_DAYS; iDate++)
{
    memcpy(&struWeekPlan.struPlanCfg[iDate][0], lpPlanSegment,
sizeof(struSinglePlanSegment));
}

BOOL bRet2 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_CARD_RIGHT_WEEK_PLAN, 1, \
&struWeekPlan, sizeof(struWeekPlan));
if (!bRet2)

```

```

{
    printf("Setting week schedule for access permission failed, error:%d.
\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set holiday group for access permission
CString m_csGroupName = "access permission holiday group 1";
NET_DVR_HOLIDAY_GROUP_CFG struHolidayGroup1 = {0};
struHolidayGroup1.dwSize = sizeof(struHolidayGroup1);
struHolidayGroup1.byEnable = 1;
strncpy((char *)struHolidayGroup1.byGroupName, (LPCTSTR)m_csGroupName,
HOLIDAY_GROUP_NAME_LEN);
struHolidayGroup1.dwHolidayPlanNo[0] = 1;//Holiday group 1 links to holiday
schedule 1,
                                         //up to 16 holiday schedules
can be linked to one holiday group

BOOL bRet3 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_CARD_RIGHT_HOLIDAY_GROUP, 1, \
    &struHolidayGroup1, sizeof(struHolidayGroup1));
if (!bRet3)
{
    printf("Setting holiday group for access permission failed, error:%d.
\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set holiday schedule for access permission
NET_DVR_HOLIDAY_PLAN_CFG struHolidayPlan = {0};
struHolidayPlan.dwSize = sizeof(struHolidayPlan);
struHolidayPlan.byEnable = 1;
struHolidayPlan.struBeginDate.wYear = 2017;//Holiday start date
struHolidayPlan.struBeginDate.byMonth = 10;
struHolidayPlan.struBeginDate.byDay = 1;
struHolidayPlan.struEndDate.wYear = 2017;//Holiday end date
struHolidayPlan.struEndDate.byMonth = 10;
struHolidayPlan.struEndDate.byDay = 7;
//Copy the week schedule parameters to holiday schedule of access permission
memcpy(struHolidayPlan.struPlanCfg, struWeekPlan.struPlanCfg,
sizeof(NET_DVR_SINGLE_PLAN_SEGMENT)*MAX_DAYS*MAX_TIMESEGMENT_V30);

BOOL bRet4 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_CARD_RIGHT_HOLIDAY_PLAN, 1, \
    &struHolidayPlan, sizeof(struHolidayPlan));
if (!bRet4)
{
    printf("Setting holiday schedule for access permission failed, error:%d.

```

```
\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
//-----
//Exit
Sleep(5000);

//Log out
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();
return;
}
```

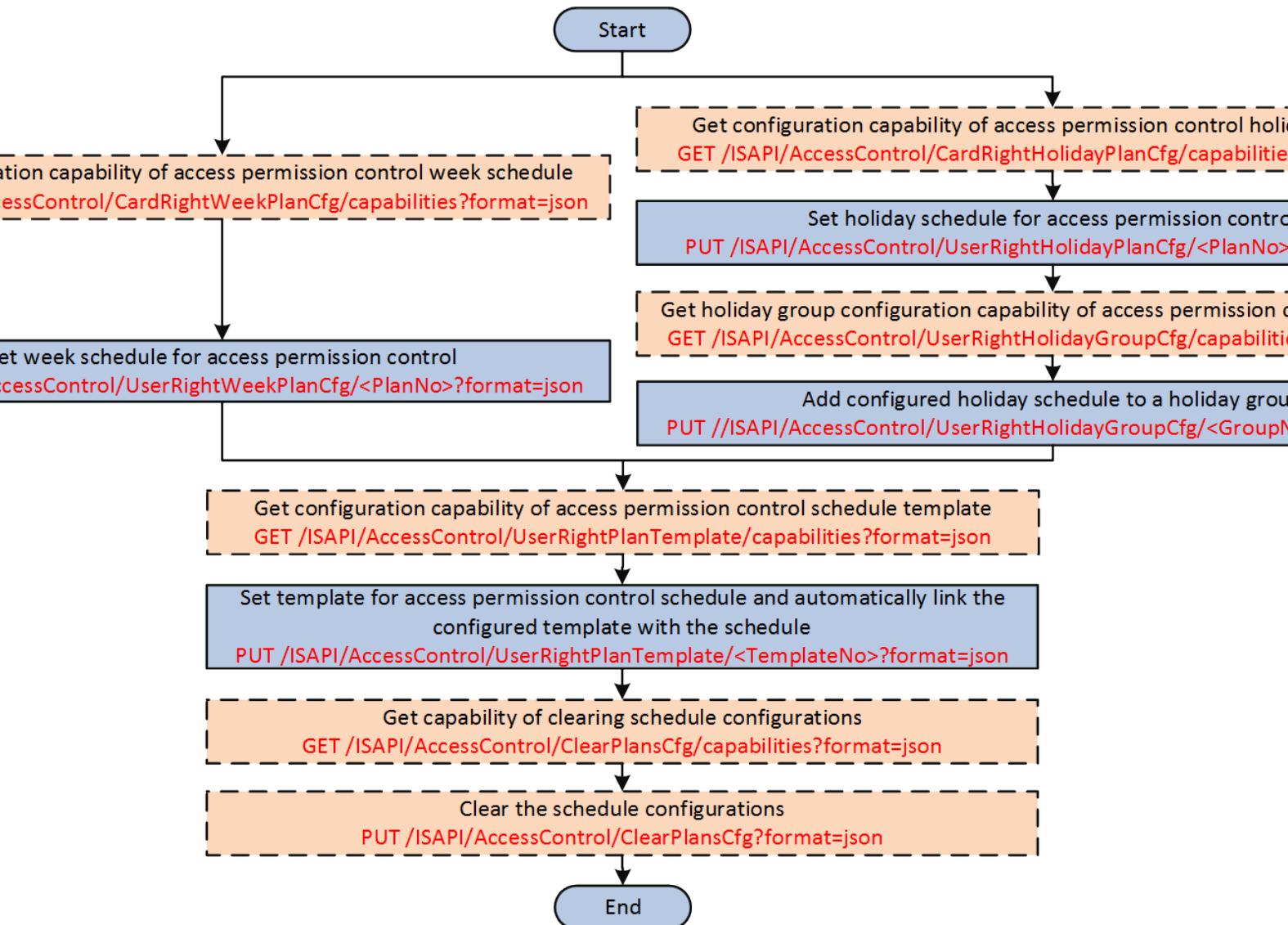
### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release the resource.

## 2.8.2 Configure Access Permission Control Schedule (Integrate by Transmitting Text Protocol)

To regularly control the access permissions for managing the accessible time duration (by default, it is 24 hours) of some important access control points, you can configure the week or holiday schedules.

## Steps



**Figure 1-15 API Calling Flow of Configuring Access Permission Control Schedule**



For integration via Device Network SDK, the related text protocol data should be transmitted by the specific API (i.e., [NET\\_DVR\\_STDXMLConfig](#)) to realize the applications. Refer to [Integrate by Transmitting Text Protocol](#) for details.

1. Perform one of the following operations to set week or holiday schedule for access permission control.

- a. Call [/ISAPI/AccessControl/UserRightWeekPlanCfg/capabilities?format=json](#) by GET method to get the configuration capability of access permission control week schedule for knowing the configuration details and notices.
  - b. Call [/ISAPI/AccessControl/UserRightWeekPlanCfg/<PlanNo>?format=json](#) by PUT method to set the week schedule.
  - a. Call [/ISAPI/AccessControl/UserRightHolidayPlanCfg/capabilities?format=json](#) by GET method to get the configuration capability of access permission control holiday schedule for knowing the configuration details and notices.
  - b. Call [/ISAPI/AccessControl/UserRightHolidayPlanCfg/<PlanNo>?format=json](#) by PUT method to set the holiday schedule.
  - c. Call [/ISAPI/AccessControl/UserRightHolidayGroupCfg/capabilities?format=json](#) by GET method to get the holiday group configuration capability of access permission control schedule for knowing the configuration details and notices.
  - d. Call [/ISAPI/AccessControl/UserRightHolidayGroupCfg/<GroupNo>?format=json](#) by PUT method to add the configured holiday schedule to a holiday group for management.
2. **Optional:** Call [/ISAPI/AccessControl/UserRightPlanTemplate/capabilities?format=json](#) by GET method to get the configuration capability of access permission control schedule template for knowing the configuration details and notices.
3. Call [/ISAPI/AccessControl/UserRightPlanTemplate/<TemplateNo>?format=json](#) by PUT method to set template for access permission control schedule and link the configured template to the schedule.



### Note

For the above configuration URLs, before setting the parameters, you'd better perform GET operation to get the existing or configured parameters for reference.

---

1. **Optional:** Call [/ISAPI/AccessControl/ClearPlansCfg/capabilities?format=json](#) by GET method to get the capability of clearing schedule configurations for knowing the configuration details and notices.
5. **Optional:** Call [/ISAPI/AccessControl/ClearPlansCfg?format=json](#) by PUT method to clear the schedule configurations.

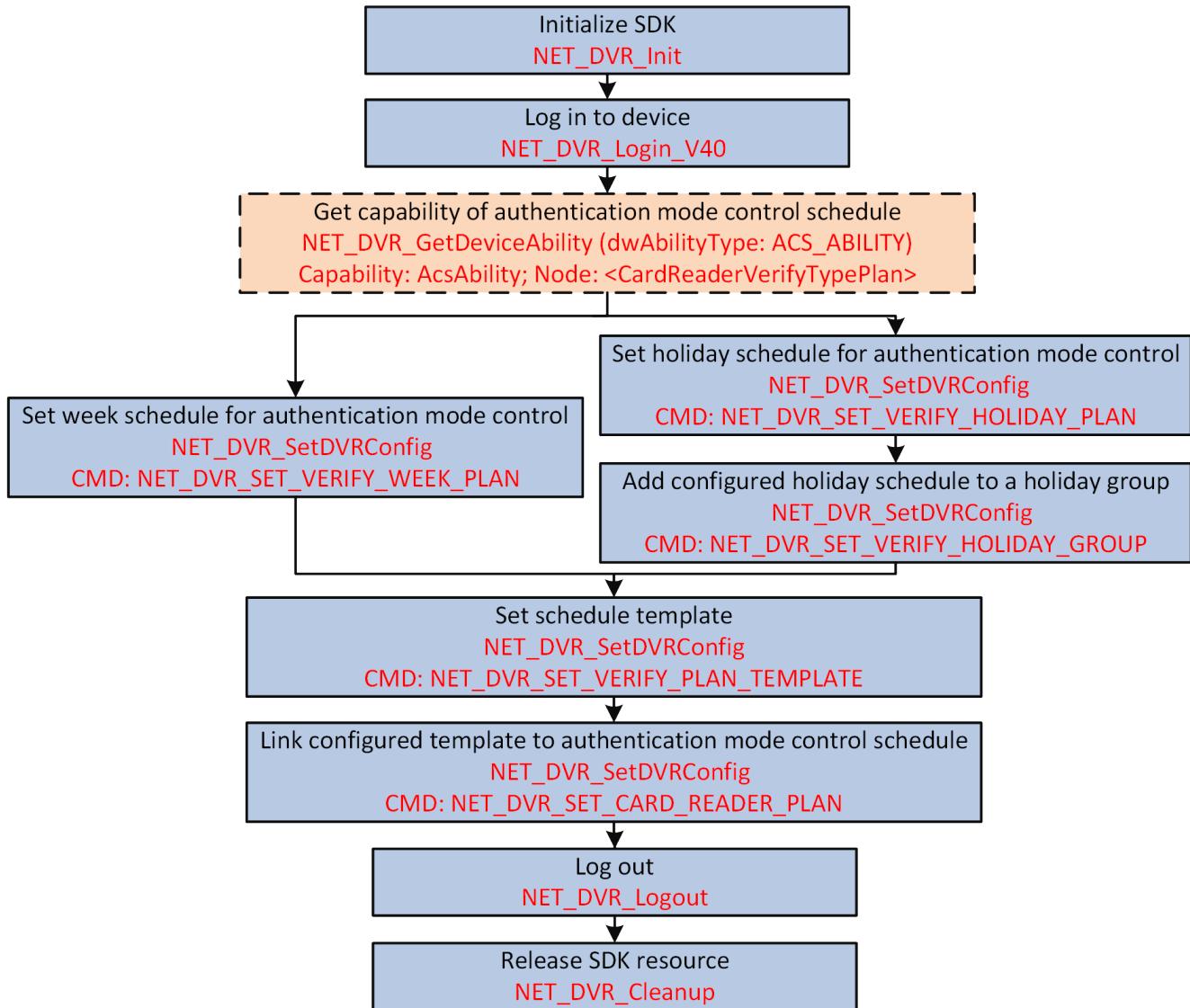
### 2.8.3 Configure Authentication Mode Control Schedule

You can configure the week or holiday schedule to regularly control the authentication modes (e.g., by card, by card+password, by fingerprint, by fingerprint+card, and so on) in some specific time periods.

#### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to device.

## Steps



**Figure 1-16 Programming Flow of Configuring Authentication Mode Control Schedule**

1. **Optional:** Call `NET_DVR_GetDeviceAbility`, specify the capability type `dwAbilityType` to "ACS\_ABILITY", set the input buffer (`pInBuf`) to `XML_Desc_AcsAbility` for getting the access control capability to check if configuring authentication mode control schedule is supported. The capability is returned in the message `XML_AcsAbility` by the output pointer (`pOutBuf`). If the node `<CardReaderVerifyTypePlan>` is returned, it indicates that configuring authentication mode control schedule is supported, and you can continue to perform the following steps. Otherwise, configuring authentication mode control schedule is not supported, please end this task.
2. Perform one of the following operations to set week or holiday schedule for authentication mode control.

- a. Call **NET\_DVR\_GetDVRConfig** with  
**NET\_DVR\_GET\_VERIFY\_WEEK\_PLAN**  
(command No.: 2124) to get the existing week schedule configurations for reference.
- 



### Note

The week schedule parameters are returned in the structure **NET\_DVR\_WEEK\_PLAN\_CFG** by output buffer (**IpOutBuffer**).

---

- b. Call **NET\_DVR\_SetDVRConfig** with  
**NET\_DVR\_SET\_VERIFY\_WEEK\_PLAN**  
(command No.: 2125) and set the input buffer (**IpInBuffer**) to **NET\_DVR\_WEEK\_PLAN\_CFG** for setting the week schedule.
  - a. Call **NET\_DVR\_GetDVRConfig** with  
**NET\_DVR\_GET\_VERIFY\_HOLIDAY\_PLAN**  
(command No.: 2128) to get the existing holiday schedule configurations for reference.
- 



### Note

The holiday schedule parameters are returned in the structure **NET\_DVR\_HOLIDAY\_PLAN\_CFG** by output buffer (**IpOutBuffer**).

---

- b. Call **NET\_DVR\_SetDVRConfig** with  
**NET\_DVR\_SET\_VERIFY\_HOLIDAY\_PLAN**  
(command No.: 2129) and set the input buffer (**IpInBuffer**) to **NET\_DVR\_HOLIDAY\_PLAN\_CFG** for setting the holiday schedule.
  - c. Call **NET\_DVR\_GetDVRConfig** with  
**NET\_DVR\_GET\_VERIFY\_HOLIDAY\_GROUP**  
(command No.: 2132) to get the existing holiday group configurations for reference.
- 



### Note

The holiday group parameters are returned in the structure **NET\_DVR\_HOLIDAY\_GROUP\_CFG** by output buffer (**IpOutBuffer**).

---

- d. Call **NET\_DVR\_SetDVRConfig** with  
**NET\_DVR\_SET\_VERIFY\_HOLIDAY\_GROUP**  
(command No.: 2133) and set the input buffer (**IpInBuffer**) to **NET\_DVR\_HOLIDAY\_GROUP\_CFG** for adding the configured holiday schedule to a holiday group.
- 

### 3. Optional: Call **NET\_DVR\_GetDVRConfig** with

**NET\_DVR\_GET\_VERIFY\_PLAN\_TEMPLATE**

(command No.: 2136) to get the existing schedule template configurations for reference.

---



The schedule template parameters are returned in the structure **NET\_DVR\_PLAN\_TEMPLATE** by output buffer (**IpOutBuffer**).

---

### 4. Call **NET\_DVR\_SetDVRConfig** with

**NET\_DVR\_SET\_VERIFY\_PLAN\_TEMPLATE**

(command No.: 2137) and set the input buffer (**lpInBuffer**) to **NET\_DVR\_PLAN\_TEMPLATE** for setting the schedule template.

### 5. Optional: Call **NET\_DVR\_GetDVRConfig** with

**NET\_DVR\_GET\_CARD\_READER\_PLAN**

(command No.: 2142) to get the existing authentication mode control schedule configurations for reference.

---



#### Note

The authentication mode control schedule parameters are returned in the structure **NET\_DVR\_CARD\_READER\_PLAN** by output buffer (**lpOutBuffer**).

---

### 6. Call **NET\_DVR\_SetDVRConfig**

**NET\_DVR\_SET\_CARD\_READER\_PLAN**

(command No.: 2143) and set the input buffer (**lpInBuffer**) to **NET\_DVR\_CARD\_READER\_PLAN** for linking the configured template to the authentication mode control schedule and finishing the configuration.

### Example

#### Sample Code for Configuring Authentication Mode Control Schedule

```
#include <stdio.h>
#include <iostream>
#include <afx.h>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

void main()
{
    //-----
    //Initialize
    NET_DVR_Init();

    //Set connection timeout and reconnection function
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    //Log in to device
    LONG lUserID;
    //Login parameters, including device IP address, user name, password, and
so on
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsynLogin = 0; //Synchronous login mode
    strcpy(struLoginInfo.sDeviceAddress, "192.168.1.64"); //Device IP address
    struLoginInfo.wPort = 8000; //Device service port number
    strcpy(struLoginInfo.sUserName, "admin"); //User name
    strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

    //Device information, output parameter
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};
```

```
lUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
if (lUserID < 0)
{
    printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//-----
//Set card reader authentication mode schedule, template 1 linked to card
reader 1
NET_DVR_CARD_READER_PLAN struCReaderPlan = {0};
struCReaderPlan.dwSize = sizeof(struCReaderPlan);
struCReaderPlan.dwTemplateNo = 1;//Schedule template 1
BOOL bRet1 = NET_DVR_SetDVRConfig(lUserID, NET_DVR_SET_CARD_READER_PLAN, 1,
\
    &struCReaderPlan, sizeof(struCReaderPlan));
if (!bRet1)
{
    printf("Setting card reader authentication mode schedule failed, error:
%d.\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set card reader authentication mode schedule template 1, template 1 links
to week schedule 1 and holiday group 1
CString m_csTemplateName = "card reader authentication mode schedule
template 1";
NET_DVR_PLAN_TEMPLATE struPlanTem = {0};
struPlanTem.dwSize = sizeof(struPlanTem);
struPlanTem.byEnable = 1;//Enable or not: 0-No, 1-Yes
strncpy((char *)struPlanTem.byTemplateName, (LPCTSTR)m_csTemplateName,
TEMPLATE_NAME_LEN);
struPlanTem.dwWeekPlanNo = 2;//Week schedule No.2
struPlanTem.dwHolidayGroupNo[0] = 2;//Holiday group No.2, up to 16 holiday
groups can be linked to each schedule

BOOL bRet2 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_VERIFY_PLAN_TEMPLATE, 1, \
    &struPlanTem, sizeof(struPlanTem));
if (!bRet2)
{
    printf("Setting card reader authentication mode schedule template
failed, error:%d.\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
```

```

//Set week schedule 2 for card reader authentication mode
NET_DVR_WEEK_PLAN_CFG struWeekPlan2 = {0};
struWeekPlan2.dwSize = sizeof(struWeekPlan2);
struWeekPlan2.byEnable = 1;//Enable week schedule

NET_DVR_SINGLE_PLAN_SEGMENT struSinglePlanSegment = {0};
LPNET_DVR_SINGLE_PLAN_SEGMENT lpPlanSegment = &struSinglePlanSegment;
struSinglePlanSegment.byEnable = 1;
struSinglePlanSegment.byVerifyMode = 4;//Authentication mode: 0-invalid, 1-
sleepy, 2-card+password, 3-card,
                                         //4-card or password, 5-
fingerprint, 6-fingerprint+password, 7-fingerprint or card,
                                         //8-fingerprint+card, 9-
fingerprint+card+password
struSinglePlanSegment.struTimeSegment.struBeginTime.byHour = 0;//Start time
struSinglePlanSegment.struTimeSegment.struBeginTime.byMinute = 0;
struSinglePlanSegment.struTimeSegment.struBeginTime.bySecond = 0;

struSinglePlanSegment.struTimeSegment.struEndTime.byHour = 23;//End time
struSinglePlanSegment.struTimeSegment.struEndTime.byMinute = 59;
struSinglePlanSegment.struTimeSegment.struEndTime.bySecond = 59;

/*Up to 8 time periods can be set for each day, and you can set different
authentication modes for each time period
Here only takes setting one period for each day*/

for (int iDate = 0; iDate<MAX_DAYS; iDate++)
{
    memcpy(&struWeekPlan2.struPlanCfg[iDate][0], lpPlanSegment,
sizeof(struSinglePlanSegment));
}

BOOL bRet3 = NET_DVR_SetDVRConfig(lUserID, NET_DVR_SET_VERIFY_WEEK_PLAN, 2,
\
                                         &struWeekPlan2, sizeof(struWeekPlan2));
if (!bRet3)
{
    printf("Setting week schedule for card reader authentication mode
failed,error:%d.\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set holiday group for card reader authentication mode
CString m_csGroupName = "Holiday group 2";
NET_DVR_HOLIDAY_GROUP_CFG struHolidayGroup2 = {0};
struHolidayGroup2.dwSize = sizeof(struHolidayGroup2);
struHolidayGroup2.byEnable = 1;
strncpy((char *)struHolidayGroup2.byGroupName, (LPCTSTR)m_csGroupName,
HOLIDAY_GROUP_NAME_LEN);
struHolidayGroup2.dwHolidayPlanNo[0] = 2;//Holiday group 1 links to holiday

```

```

schedule 1,                                     //up to 16 holiday schedules
can be linked to one holiday group

    BOOL bRet4 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_VERIFY_HOLIDAY_GROUP, 2, \
    &struHolidayGroup2, sizeof(struHolidayGroup2));
if (!bRet4)
{
    printf("Setting holiday group for card reader authentication mode
failed, error:%d.\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set holiday schedule for card reader authentication mode
NET_DVR_HOLIDAY_PLAN_CFG struHolidayPlan2 = {0};
struHolidayPlan2.dwSize = sizeof(struHolidayPlan2);
struHolidayPlan2.bEnable = 1;
struHolidayPlan2.struBeginDate.wYear = 2017;//Holiday start date
struHolidayPlan2.struBeginDate.byMonth = 10;
struHolidayPlan2.struBeginDate.byDay = 1;
struHolidayPlan2.struEndDate.wYear = 2017;//Holiday end date
struHolidayPlan2.struEndDate.byMonth = 10;
struHolidayPlan2.struEndDate.byDay = 7;
//Copy the week schedule parameters to holiday schedule of card reader
authentication mode
memcpy(struHolidayPlan2.struPlanCfg, struWeekPlan2.struPlanCfg,
sizeof(NET_DVR_SINGLE_PLAN_SEGMENT)*MAX_DAYS*MAX_TIMESEGMENT_V30);

    BOOL bRet5 = NET_DVR_SetDVRConfig(lUserID, NET_DVR_SET_VERIFY_HOLIDAY_PLAN,
2, \
    &struHolidayPlan2, sizeof(struHolidayPlan2));
if (!bRet5)
{
    printf("Setting holiday schedule for card reader authentication mode
failed, error:%d.\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
//-----
//Exit
Sleep(5000);

//Log out
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();
return;
}

```

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release the resource.

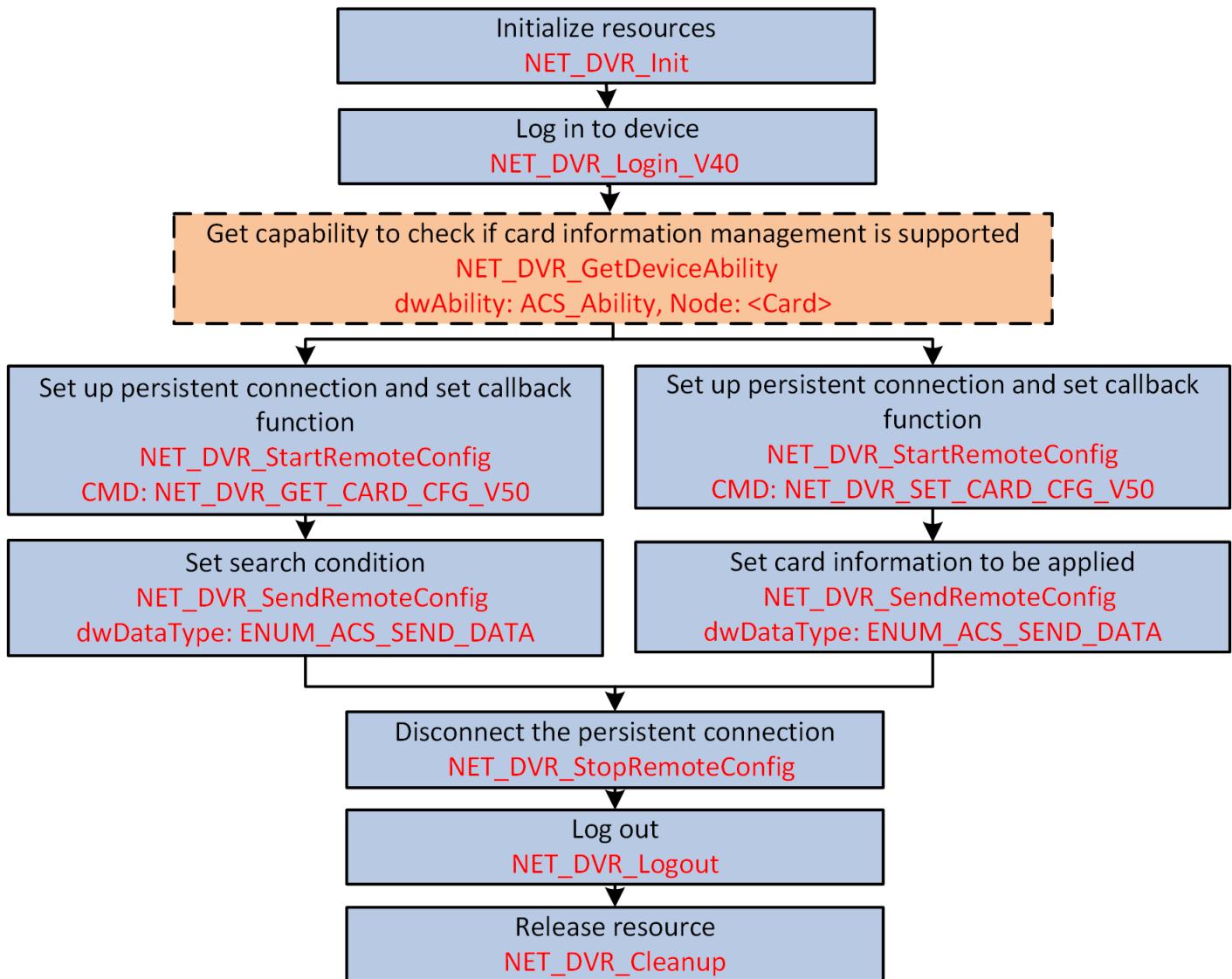
### 2.8.4 Configure Door Control Schedule

You can configure the week or holiday schedule to regularly control the door statuses, including Remain Open (access without authentication), Remain Closed (access is not allowed), and Normal (access with authentication), in some specific time periods.

#### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to device.

## Steps



**Figure 1-17 Programming Flow of Configuring Door Control Schedule**

1. Call **NET\_DVR\_GetDeviceAbility**, specify the capability type **dwAbilityType** to "ACS\_ABILITY", set the input buffer (**pInBuf**) to **XML\_Desc\_AcsAbility** for getting the access control capability to check if configuring door control schedule is supported.  
The capability is returned in the message **XML\_AcsAbility** by the output pointer (**pOutBuf**).  
If the node **<DoorStatusPlan>** is returned, it indicates that configuring door control schedule is supported, and you can continue to perform the following steps.  
Otherwise, configuring door control schedule is not supported, please end this task.
2. Perform one of the following operations to set week or holiday schedule for door control.
  - a. Call **NET\_DVR\_GetDVRConfig** with "NET\_DVR\_GET\_WEEK\_PLAN\_CFG" (command No.: 2100) to get the existing week schedule configurations for reference.



### Note

The week schedule parameters are returned in the structure [NET\\_DVR\\_WEEK\\_PLAN\\_CFG](#) by output buffer ([IpOutBuffer](#)).

- b. Call [NET\\_DVR\\_SetDVRConfig](#) with "NET\_DVR\_SET\_WEEK\_PLAN\_CFG" (command No.: 2101) and set the input buffer ([IpInBuffer](#)) to [NET\\_DVR\\_WEEK\\_PLAN\\_CFG](#) for setting the week schedule.
  - a. Call [NET\\_DVR\\_GetDVRConfig](#) with "NET\_DVR\_GET\_DOOR\_STATUS\_HOLIDAY\_PLAN" (command No.: 2102) to get the existing holiday schedule configurations for reference.
- 



### Note

The holiday schedule parameters are returned in the structure

[NET\\_DVR\\_HOLIDAY\\_PLAN\\_CFG](#) by output buffer ([IpOutBuffer](#)).

- b. Call [NET\\_DVR\\_SetDVRConfig](#) with "NET\_DVR\_SET\_DOOR\_STATUS\_HOLIDAY\_PLAN" (command No.: 2103) and set the input buffer ([IpInBuffer](#)) to [NET\\_DVR\\_HOLIDAY\\_PLAN\\_CFG](#) for setting the week schedule.
  - c. Call [NET\\_DVR\\_GetDVRConfig](#) with "NET\_DVR\_GET\_DOOR\_STATUS\_HOLIDAY\_GROUP" (command No.: 2104) to get the existing holiday group configurations for reference.
- 



### Note

The holiday group parameters are returned in the structure

[NET\\_DVR\\_HOLIDAY\\_GROUP\\_CFG](#) by output buffer ([IpOutBuffer](#)).

- d. Call [NET\\_DVR\\_SetDVRConfig](#) with "NET\_DVR\_SET\_DOOR\_STATUS\_HOLIDAY\_GROUP" (command No.: 2105) and set the input buffer ([IpInBuffer](#)) to [NET\\_DVR\\_HOLIDAY\\_GROUP\\_CFG](#) for adding the configured holiday schedule to a holiday group.
- 

- 3. Optional:** Call [NET\\_DVR\\_GetDVRConfig](#) with "NET\_DVR\_GET\_DOOR\_STATUS\_PLAN\_TEMPLATE" (command No.: 2106) to get the existing schedule template configurations for reference.
- 



### Note

The schedule template parameters are returned in the structure [NET\\_DVR\\_PLAN\\_TEMPLATE](#) by output buffer ([IpOutBuffer](#)).

- 4. Call [NET\\_DVR\\_SetDVRConfig](#) with "NET\_DVR\_SET\_DOOR\_STATUS\_PLAN\_TEMPLATE" (command No.: 2107) and set the input buffer ([IpInBuffer](#)) to [NET\\_DVR\\_PLAN\\_TEMPLATE](#) for setting the schedule template.
  - 5. Optional:** Call [NET\\_DVR\\_GetDVRConfig](#) with "NET\_DVR\_GET\_DOOR\_STATUS\_PLAN" (command No.: 2110) to get the existing door control schedule configurations for reference.
- 



### Note

The door control schedule parameters are returned in the structure

[NET\\_DVR\\_DOOR\\_STATUS\\_PLAN](#) by output buffer ([IpOutBuffer](#)).

---

6. Call **NET\_DVR\_SetDVRConfig** with "NET\_DVR\_SET\_DOOR\_STATUS\_PLAN" (command No.: 2111) and set the input buffer (**lpInBuffer**) to **NET\_DVR\_DOOR\_STATUS\_PLAN** for linking the configured template to the door control schedule and finishing the configuration.

### Example

#### Sample Code for Configuring Door Control Schedule

```
#include <stdio.h>
#include <iostream>
#include <afx.h>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

void main()
{
    //-----
    //Initialize
    NET_DVR_Init();

    //Set connection timeout and reconnection function
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    //Log in to device
    LONG lUserID;
    //Login parameters, including device IP address, user name, password, and
so on
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsynLogin = 0; //Synchronous login mode
    strcpy(struLoginInfo.sDeviceAddress, "192.168.1.64"); //Device IP address
    struLoginInfo.wPort = 8000; //Device service port number
    strcpy(struLoginInfo.sUserName, "admin"); //User name
    strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

    //Device information, output parameter
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

    lUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
    if (lUserID < 0)
    {
        printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

    //-----
    //Set door status schedule, template 1 linked to door 1
    NET_DVR_DOOR_STATUS_PLAN struDoorStatusPlan = {0};
    struDoorStatusPlan.dwSize = sizeof(struDoorStatusPlan);
    struDoorStatusPlan.dwTemplateNo = 1;//Schedule template 1
```

```

BOOL bRet1 = NET_DVR_SetDVRConfig(lUserID, NET_DVR_SET_DOOR_STATUS_PLAN, 1,
\
    &struDoorStatusPlan, sizeof(struDoorStatusPlan));
if (!bRet1)
{
    printf("Setting door status schedule failed, error:%d.\n",
NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set door status schedule template 1, template 1 links to week schedule 1
and holiday group 1
CString m_csTemplateName = "door status schedule template 1";
NET_DVR_PLAN_TEMPLATE struPlanTem = {0};
struPlanTem.dwSize = sizeof(struPlanTem);
struPlanTem.byEnable = 1;//Enable or not: 0-No, 1-Yes
strncpy((char *)struPlanTem.byTemplateName, (LPCTSTR)m_csTemplateName,
TEMPLATE_NAME_LEN);
struPlanTem.dwWeekPlanNo = 1;//Week schedule No.1
struPlanTem.dwHolidayGroupNo[0] = 1;//Holiday group No.1, up to 16 holiday
groups can be linked to each schedule
//struPlanTem.dwHolidayGroupNo[1] = 2;//Holiday group No.2

BOOL bRet2 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_DOOR_STATUS_PLAN_TEMPLATE, 1, \
    &struPlanTem, sizeof(struPlanTem));
if (!bRet2)
{
    printf("Setting door status schedule template failed, error:%d.\n",
NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set week schedule 1 for door status
NET_DVR_WEEK_PLAN_CFG struWeekPlan = {0};
struWeekPlan.dwSize = sizeof(struWeekPlan);
struWeekPlan.byEnable = 1;//Enable week scheudle

NET_DVR_SINGLE_PLAN_SEGMENT struSinglePlanSegment = {0};
LPNET_DVR_SINGLE_PLAN_SEGMENT lpPlanSegment = &struSinglePlanSegment;
struSinglePlanSegment.byEnable = 1;
struSinglePlanSegment.byDoorStatus = 3;//Door status: 0-invalid, 1-sleepy,
2-remain open, 3-remain closed.
struSinglePlanSegment.struTimeSegment.struBeginTime.byHour = 0;//Start time
struSinglePlanSegment.struTimeSegment.struBeginTime.byMinute = 0;
struSinglePlanSegment.struTimeSegment.struBeginTime.bySecond = 0;

```

```

struSinglePlanSegment.struTimeSegment.struEndTime.byHour = 23;//End time
struSinglePlanSegment.struTimeSegment.struEndTime.byMinute = 59;
struSinglePlanSegment.struTimeSegment.struEndTime.bySecond = 59;

/*Up to 8 time periods can be set for each day, and you can set different
statuses for each time period
Here only takes setting one period for each day*/

for (int iDate = 0; iDate<MAX_DAYS; iDate++)
{
    memcpy(&struWeekPlan.struPlanCfg[iDate][0], lpPlanSegment,
sizeof(struSinglePlanSegment));
}

BOOL bRet3 = NET_DVR_SetDVRConfig(lUserID, NET_DVR_SET_WEEK_PLAN_CFG, 1, \
&struWeekPlan, sizeof(struWeekPlan));
if (!bRet3)
{
    printf("Setting week schedule for door status failed, error:%d.\n",
NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set holiday group for door status
CString m_csGroupName = "door status holiday group 1";
NET_DVR_HOLIDAY_GROUP_CFG struHolidayGroup1 = {0};
struHolidayGroup1.dwSize = sizeof(struHolidayGroup1);
struHolidayGroup1.byEnable = 1;
strncpy((char *)struHolidayGroup1.byGroupName, (LPCTSTR)m_csGroupName,
HOLIDAY_GROUP_NAME_LEN);
struHolidayGroup1.dwHolidayPlanNo[0] = 1;//Holiday group 1 links to holiday
schedule 1,                                              //up to 16 holiday schedules
can be linked to one holiday group

BOOL bRet4 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_DOOR_STATUS_HOLIDAY_GROUP, 1, \
&struHolidayGroup1, sizeof(struHolidayGroup1));
if (!bRet4)
{
    printf("Setting holiday group for door status failed, error:%d.\n",
NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Set holiday schedule for door status
NET_DVR_HOLIDAY_PLAN_CFG struHolidayPlan = {0};
struHolidayPlan.dwSize = sizeof(struHolidayPlan);

```

```
struHolidayPlan.byEnable = 1;
struHolidayPlan.struBeginDate.wYear = 2017;//Holiday start date
struHolidayPlan.struBeginDate.byMonth = 10;
struHolidayPlan.struBeginDate.byDay = 1;
struHolidayPlan.struEndDate.wYear = 2017;//Holiday end date
struHolidayPlan.struEndDate.byMonth = 10;
struHolidayPlan.struEndDate.byDay = 7;
//Copy the week schedule parameters to holiday schedule of door status
memcpy(struHolidayPlan.struPlanCfg, struWeekPlan.struPlanCfg,
sizeof(NET_DVR_SINGLE_PLAN_SEGMENT)*MAX_DAYS*MAX_TIMESEGMENT_V30);

BOOL bRet5 = NET_DVR_SetDVRConfig(lUserID,
NET_DVR_SET_DOOR_STATUS_HOLIDAY_PLAN, 1, \
&struHolidayPlan, sizeof(struHolidayPlan));
if (!bRet5)
{
    printf("Setting holiday schedule for door status failed, error:%d.\n",
NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}
//-----
//Exit
Sleep(5000);

//Log out
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();
return;
}
```

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release the resource.

## 2.9 Alarm and Event Receiving

The alarm/event information from the device can be received in third-party platform or system when the alarms are triggered or event occurred. Two modes are available for receiving alarms, including arming mode and listening mode.

### Arming Mode

The third-party platform connects to device automatically, when the alarm is triggered, the platform sends alarm uploading command to the device, and then the device will upload the alarm to the platform.

### Listening Mode

When alarm is triggered, the device automatically uploads the alarm, and then the third-party platform receives the uploaded alarm via the configured listening host (listening address and port should be configured). This mode is applicable for multiple devices uploading alarm/event information to one third-party platform without logging in to devices, and the restart of devices will not affect the alarm/event uploading. But a device can only support the configuration of one or two listening addresses and ports.

### 2.9.1 Access Control Event Types

The access control events are classified as four major types, i.e., alarm events (MAJOR\_ALARM-0x1), exception events (MAJOR\_EXCEPTION-0x2), operation events (MAJOR\_OPERATION-0x3), and other events (MAJOR\_EVENT-0x5). Each major type corresponds to multiple minor types, see details below.

#### MAJOR\_ALARM

Event Minor Type	Value	Description
MINOR_ALARMIN_SHORT_CIRCUIT	0x400	Zone Short Circuit Attempts Alarm
MINOR_ALARMIN_BROKEN_CIRCUIT	0x401	Zone Disconnected Alarm
MINOR_ALARMIN_EXCEPTION	0x402	Zone Exception Alarm
MINOR_ALARMIN_RESUME	0x403	Zone Restored
MINOR_HOST_DESMANTLE_ALARM	0x404	Zone Tampering Alarm
MINOR_HOST_DESMANTLE_RESUME	0x405	Zone Tampering Restored
MINOR_CARD_READER_DESMANTLE_ALARM	0x406	Card Reader Tampering Alarm
MINOR_CARD_READER_DESMANTLE_RESUME	0x407	Card Reader Tampering Restored
MINOR_CASE_SENSOR_ALARM	0x408	Alarm Input Alarm Triggered
MINOR_CASE_SENSOR_RESUME	0x409	Alarm Input Restored
MINOR_STRESS_ALARM	0x40a	Duress Alarm
MINOR_OFFLINE_ECENT_NEARLY_FULL	0x40b	No Memory Alarm for Offline Events

Event Minor Type	Value	Description
MINOR_CARD_MAX_AUTHENTICATE_FAIL	0x40c	Maximum Failed Card Authentications Alarm
MINOR_SD_CARD_FULL	0x40d	SD Card Full Alarm
MINOR_LINKAGE_CAPTURE_PIC	0x40e	Capture Linkage Alarm
MINOR_SECURITY_MODULE_DESMANTLE_ALARM	0x40f	Secure Door Control Unit Tampering Alarm
MINOR_SECURITY_MODULE_DESMANTLE_RESUME	0x410	Secure Door Control Unit Tampering Restored
MINOR_FIRE_IMPORT_SHORT_CIRCUIT	0x415	Fire Input Short Circuit Attempts Alarm
MINOR_FIRE_IMPORT_BROKEN_CIRCUIT	0x416	Fire Input Open Circuit Attempts Alarm
MINOR_FIRE_IMPORT_RESUME	0x417	Fire Input Restored
MINOR_FIRE_BUTTON_TRIGGER	0x418	Fire Button Triggered
MINOR_FIRE_BUTTON_RESUME	0x419	Fire Button Resumed
MINOR_MAINTENANCE_BUTTON_TRIGGER	0x41a	Maintenance Button Triggered
MINOR_MAINTENANCE_BUTTON_RESUME	0x41b	Maintenance Button Resumed
MINOR_EMERGENCY_BUTTON_TRIGGER	0x41c	Panic Button Triggered
MINOR_EMERGENCY_BUTTON_RESUME	0x41d	Panic Button Resumed
MINOR_DISTRACT_CONTROLLER_ALARM	0x41e	Distributed Elevator Controller Tampering Alarm
MINOR_DISTRACT_CONTROLLER_RESUME	0x41f	Distributed Elevator Controller Tampering Restored

Event Minor Type	Value	Description
MINOR_CHANNEL_CONTROLLER_DESMANTLE_ALARM	0x422	Lane Controller Tampering Alarm
MINOR_CHANNEL_CONTROLLER_DESMANTLE_RESUME	0x423	Lane Controller Tampering Alarm Restored
MINOR_CHANNEL_CONTROLLER_FIRE_IMPORT_ALARM	0x424	Lane Controller Fire Input Alarm
MINOR_CHANNEL_CONTROLLER_FIRE_IMPORT_RESUME	0x425	Lane Controller Fire Input Alarm Restored
MINOR_PRINTER_OUT_OF_PAPER	0x440	No Paper in Printer Alarm
MINOR_LEGAL_EVENT_NEARLY_FULL	0x442	No Memory Alarm for Valid Offline Events
MINOR_ALARM_CUSTOM1 to MINOR_ALARM_CUSTOM64	0x900 to 0x93f	Access Control: Custom Alarm Event 1 to Custom Alarm Event 64

## MAJOR\_EXCEPTION

Event Minor Type	Value	Description
MINOR_NET_BROKEN	0x27	Network Disconnected
MINOR_RS485_DEVICE_ABNORMAL	0x3a	RS485 Connection Exception
MINOR_RS485_DEVICE_REVERT	0x3b	RS485 Connection Restored
MINOR_DEV_POWER_ON	0x400	Power on
MINOR_DEV_POWER_OFF	0x401	Power off
MINOR_WATCH_DOG_RESET	0x402	Watchdog Reset
MINOR_LOW_BATTERY	0x403	Low Battery Voltage
MINOR_BATTERY_RESUME	0x404	Battery Voltage Restored

Event Minor Type	Value	Description
MINOR_AC_OFF	0x405	AC Power Disconnected
MINOR_AC_RESUME	0x406	AC Power Restored
MINOR_NET_RESUME	0x407	Network Restored
MINOR_FLASH_ABNORMAL	0x408	Flash Reading and Writing Exception
MINOR_CARD_READER_OFFLINE	0x409	Card Reader Offline
MINOR_CAED_READER_RESUME	0x40a	Card Reader Online
MINOR_INDICATOR_LIGHT_OFF	0x40b	Indicator Turns off
MINOR_INDICATOR_LIGHT_RESUME	0x40c	Indicator Resumed
MINOR_CHANNEL_CONTROLLER_OFF	0x40d	Lane Controller Offline
MINOR_CHANNEL_CONTROLLER_RESUME	0x40e	Lane Controller Online
MINOR_SECURITY_MODULE_OFF	0x40f	Secure Door Control Unit Offline
MINOR_SECURITY_MODULE_RESUME	0x410	Secure Door Control Unit Online
MINOR_BATTERY_ELECTRIC_LOW	0x411	Low Battery Voltage (Only for Face Recognition Terminal)
MINOR_BATTERY_ELECTRIC_RESUME	0x412	Battery Voltage Recovered (Only for Face Recognition Terminal)
MINOR_LOCAL_CONTROL_NET_BROKEN	0x413	Network of Distributed Access Controller Disconnected
MINOR_LOCAL_CONTROL_NET_RSUME	0x414	Network of Distributed Access Controller Restored
MINOR_MASTER_RS485_LOOPNODE_BROKEN	0x415	RS485 Loop of Main Access Controller Disconnected

Event Minor Type	Value	Description
MINOR_MASTER_RS485_LOOPNODE_RESUME	0x416	RS485 Loop of Main Access Controller Connected
MINOR_LOCAL_CONTROL_OFFLINE	0x417	Distributed Access Controller Offline
MINOR_LOCAL_CONTROL_RESUME	0x418	Distributed Access Controller Online
MINOR_LOCAL_DOWNSIDE_RS485_LOOPNODE_BROKEN	0x419	Downstream RS485 Loop of Distributed Access Control Disconnected
MINOR_LOCAL_DOWNSIDE_RS485_LOOPNODE_RESUME	0x41a	Downstream RS485 Loop of Distributed Access Control Connected
MINOR_DISTRACT_CONTROLLER_ONLINE	0x41b	Distributed Elevator Controller Online
MINOR_DISTRACT_CONTROLLER_OFFLINE	0x41c	Distributed Elevator Controller Offline
MINOR_ID_CARD_READER_NOT_CONNECT	0x41d	ID Card Reader Disconnected
MINOR_ID_CARD_READER_RESUME	0x41e	ID Card Reader Connected
MINOR_FINGER_PRINT_MODULE_NOT_CONNECT	0x41f	Fingerprint Module Disconnected
MINOR_FINGER_PRINT_MODULE_RESUME	0x420	Fingerprint Module Connected
MINOR_CAMERA_NOT_CONNECT	0x421	Camera Disconnected
MINOR_CAMERA_RESUME	0x422	Camera Connected
MINOR_COM_NOT_CONNECT	0x423	COM Port Disconnected
MINOR_COM_RESUME	0x424	COM Port Connected
MINOR_DEVICE_NOT_AUTHORIZE	0x425	Device Unauthorized
MINOR_PEOPLE_AND_ID_CARD_DEVICE_ONLINE	0x426	Face Recognition Terminal Online

Event Minor Type	Value	Description
MINOR_PEOPLE_AND_ID_CARD_DEVICE_OFFLINE	0x427	Face Recognition Terminal Offline
MINOR_LOCAL_LOGIN_LOCK	0x428	Local Login Lock
MINOR_LOCAL_LOGIN_UNLOCK	0x429	Local Login Unlock
MINOR_SUBMARINEBACK_COMM_BREAK	0x42a	Communication with Anti-passing Back Server Failed
MINOR_SUBMARINEBACK_COMM_RESUME	0x42b	Communication with Anti-passing Back Server Restored
MINOR_MOTOR_SENSOR_EXCEPTION	0x42c	Motor or Sensor Exception
MINOR_CAN_BUS_EXCEPTION	0x42d	CAN Bus Exception
MINOR_CAN_BUS_RESUME	0x42e	CAN Bus Exception Restored
MINOR_GATE_TEMPERATURE_OVERRUN	0x42f	Too High Pedestal Temperature
MINOR_IR_EMITTER_EXCEPTION	0x430	Active Infrared Intrusion Detector Exception
MINOR_IR_EMITTER_RESUME	0x431	Active Infrared Intrusion Detector Restored
MINOR_LAMP_BOARD_COMM_EXCEPTION	0x432	Communication with Light Board Failed
MINOR_LAMP_BOARD_COMM_RESUME	0x433	Communication with Light Board Restored
MINOR_IR_ADAPTOR_COMM_EXCEPTION	0x434	Communication with IR Adaptor Failed
MINOR_IR_ADAPTOR_COMM_RESUME	0x435	Communication with IR Adaptor Restored
MINOR_PRINTER_ONLINE	0x436	Printer Online
MINOR_PRINTER_OFFLINE	0x437	Printer Offline
MINOR_4G_MOUDLE_ONLINE	0x438	4G Module Online
MINOR_4G_MOUDLE_OFFLINE	0x439	4G Module Offline

Event Minor Type	Value	Description
MINOR_AUXILIARY_BOARD_OFFLINE	0x43c	Auxiliary Board Disconnected
MINOR_AUXILIARY_BOARD_RESUME	0x43d	Auxiliary Board Connected
MINOR_IDCARD_SECURITY_MOUDLE_EXCEPTION	0x43e	Secure ID Card Unit Exception
MINOR_IDCARD_SECURITY_MOUDLE_RESUME	0x43f	Secure ID Card Unit Restored
MINOR_FP_PERIPHERAL_EXCEPTION	0x440	Fingerprint Collection Peripheral Exception
MINOR_FP_PERIPHERAL_RESUME	0x441	Fingerprint Collection Peripheral Restored
MINOR_EXTEND_MODULE_ONLINE	0x44d	Extension Module Online
MINOR_EXTEND_MODULE_OFFLINE	0x44e	Extension Module Offline
MINOR_EXCEPTION_CUSTOM1 to MINOR_EXCEPTION_CUSTOM64	0x900 to 0x93f	Access Control: Custom Exception Event 1 to Custom Exception Event 64

## MAJOR\_OPERATION

Alarm Minor Types	Value	Description
MINOR_LOCAL_LOGIN	0x50	Local Login
MINOR_LOCAL_LOGOUT	0x51	Local Logout
MINOR_LOCAL_UPGRADE	0x5a	Local Upgrade
MINOR_REMOTE_LOGIN	0x70	Remote Login
MINOR_REMOTE_LOGOUT	0x71	Remote Logout
MINOR_REMOTE_ARM	0x79	Remote Arming
MINOR_REMOTE_DISARM	0x7a	Remote Disarming
MINOR_REMOTE_REBOOT	0x7b	Remote Reboot
MINOR_REMOTE_UPGRADE	0x7e	Remote Upgrade

Alarm Minor Types	Value	Description
MINOR_REMOTE_CFGFILE_OUTPUT	0x86	Remote Operation: Export Configuration File
MINOR_REMOTE_CFGFILE_INP	0x87	Remote Operation: Import Configuration File
MINOR_REMOTE_ALARMOUT_OPEN_MAN	0xd6	Remote Operation: Enable Alarm Output Manually
MINOR_REMOTE_ALARMOUT_CLOSE_MAN	0xd7	Remote Operation: Disable Alarm Output Manually
MINOR_REMOTE_OPEN_DOOR	0x400	Door Remotely Open
MINOR_REMOTE_CLOSE_DOOR	0x401	Door Remotely Closed
MINOR_REMOTE_ALWAYS_OPEN	0x402	Remain Open Remotely
MINOR_REMOTE_ALWAYS_CLOSE	0x403	Remain Closed Remotely
MINOR_REMOTE_CHECK_TIME	0x404	Remote: Manual Time Sync
MINOR_NTP_CHECK_TIME	0x405	Network Time Protocol Synchronization
MINOR_REMOTE_CLEAR_CARD	0x406	Remote Operation: Clear All Card No.
MINOR_REMOTE_RESTORE_CFG	0x407	Remote Operation: Restore Defaults
MINOR_ALARMIN_ARM	0x408	Zone Arming
MINOR_ALARMIN_DISARM	0x409	Zone Disarming
MINOR_LOCAL_RESTORE_CFG	0x40a	Local Operation: Restore Defaults
MINOR_REMOTE_CAPTURE_PIC	0x40b	Remote Operation: Capture
MINOR_MOD_NET_REPORT_CFG	0x40c	Edit Network Parameters
MINOR_MOD_GPRS_REPORT_PARAM	0x40d	Edit GPRS Parameters

Alarm Minor Types	Value	Description
MINOR_MOD_REPORT_GROUP_PARAM	0x40e	Edit Control Center Parameters
MINOR_UNLOCK_PASSWORD_OPEN_DOOR	0x40f	Enter Dismiss Code
MINOR_AUTO_RENUMBER	0x410	Auto Renumber
MINOR_AUTO_COMPLEMENT_NUMBER	0x411	Auto Supplement Number
MINOR_NORMAL_CFGFILE_INPUT	0x412	Import Configuration File
MINOR_NORMAL_CFGFILE_OUTPUT	0x413	Export Configuration File
MINOR_CARD_RIGHT_INPUT	0x414	Import Card Permission Parameters
MINOR_CARD_RIGHT_OUTPUT	0x415	Export Card Permission Parameters
MINOR_LOCAL_USB_UPGRADE	0x416	Upgrade Device via USB flash Drive
MINOR_REMOTE_VISITOR_CALL_LADDER	0x417	Visitor Calling Elevator
MINOR_REMOTE_HOUSEHOLD_CALL_LADDER	0x418	Resident Calling Elevator
MINOR_REMOTE_ACTUAL_GUARD	0x419	Remotely Arming
MINOR_REMOTE_ACTUAL_UNGUARD	0x41a	Remotely Disarming
MINOR_REMOTE_CONTROL_NOT_CODE_OPER_FAILED	0x41b	Operation Failed: Keyfob Not Pairing
MINOR_REMOTE_CONTROL_CLOSE_DOOR	0x41c	Keyfob Operation: Close Door
MINOR_REMOTE_CONTROL_OPEN_DOOR	0x41d	Keyfob Operation: Open Door
MINOR_REMOTE_CONTROL_ALWAYS_OPEN_DOOR	0x41e	Keyfob Operation: Remain Door Open

Alarm Minor Types	Value	Description
MINOR_M1_CARD_ENCRYPT_VERIFY_OPEN	0x41f	M1 Card Encryption Verification Enabled
MINOR_M1_CARD_ENCRYPT_VERIFY_CLOSE	0x420	M1 Card Encryption Verification Disabled
MINOR_NFC_FUNCTION_OPEN	0X421	Opening Door with NFC Card Enabled
MINOR_NFC_FUNCTION_CLOSE	0X422	Opening Door with NFC Card Disabled
MINOR_OFFLINE_DATA_OUTPUT	0x423	Export Offline Collected Data
MINOR_CREATE_SSH_LINK	0x42d	Establish SSH Connection
MINOR_CLOSE_SSH_LINK	0x42e	Disconnect SSH Connection
MINOR_BLUETOOTH_KEY MODIFY	/	Bluetooth Key Modified
MINOR_OPERATION_CUSTOM1 to MINOR_OPERATION_CUSTOM64	0x900-0x93f	Access Control: Custom Operation Event 1 to Custom Operation Event 64

## MAJOR\_EVENT

Event Minor Types	Value	Description
MINOR_LEGAL_CARD_PASS	0x01	Valid Card Authentication Completed
MINOR_CARD_AND_PSW_PASS	0x02	Card and Password Authentication Completed
MINOR_CARD_AND_PSW_FAIL	0x03	Card and Password Authentication Failed
MINOR_CARD_AND_PSW_TIMEOUT	0x04	Card and Password Authentication Timed Out
MINOR_CARD_AND_PSW_OVER_TIME	0x05	Card and Password Authentication Timed Out
MINOR_CARD_NO_RIGHT	0x06	No Permission

Event Minor Types	Value	Description
MINOR_CARD_INVALID_PERIOD	0x07	Invalid Card Swiping Time Period
MINOR_CARD_OUT_OF_DATE	0x08	Expired Card
MINOR_INVALID_CARD	0x09	Card No. Not Exist
MINOR_ANTI_SNEAK_FAIL	0x0a	Anti-passing Back Authentication Failed
MINOR_INTERLOCK_DOOR_NOT_CLOSE	0x0b	Interlocking Door Not Closed
MINOR_NOT_BELONG_MULTI_GROUP	0x0c	Card Not in Multiple Authentication Group
MINOR_INVALID_MULTI_VERIFY_PERIOD	0x0d	Card Not in Multiple Authentication Duration
MINOR_MULTI_VERIFY_SUPER_RIGHT_FAIL	0x0e	Multiple Authentications: Super Password Authentication Failed
MINOR_MULTI_VERIFY_REMOTE_RIGHT_FAIL	0x0f	Multiple Authentication Completed
MINOR_MULTI_VERIFY_SUCCESS	0x10	Multiple Authenticated
MINOR_LEADER_CARD_OPEN_BEGIN	0x11	Open Door with First Card Started
MINOR_LEADER_CARD_OPEN_END	0x12	Open Door with First Card Stopped
MINOR_ALWAYS_OPEN_BEGIN	0x13	Remain Open Started
MINOR_ALWAYS_OPEN_END	0x14	Remain Open Stopped
MINOR_LOCK_OPEN	0x15	Door Unlocked
MINOR_LOCK_CLOSE	0x16	Door Locked
MINOR_DOOR_BUTTON_PRESS	0x17	Exit Button Pressed
MINOR_DOOR_BUTTON_RELEASE	0x18	Exit Button Released
MINOR_DOOR_OPEN_NORMAL	0x19	Door Open (Contact)

Event Minor Types	Value	Description
MINOR_DOOR_CLOSE_NORMAL	0x1a	Door Closed (Contact)
MINOR_DOOR_OPEN_ABNORMAL	0x1b	Door Abnormally Open (Contact)
MINOR_DOOR_OPEN_TIMEOUT	0x1c	Door Open Timed Out (Contact)
MINOR_ALARMOUT_ON	0x1d	Alarm Output Enabled
MINOR_ALARMOUT_OFF	0x1e	Alarm Output Disabled
MINOR_ALWAYS_CLOSE_BEGIN	0x1f	Remain Closed Started
MINOR_ALWAYS_CLOSE_END	0x20	Remain Closed Stopped
MINOR_MULTI_VERIFY_NEED_REMOTE_OPEN	0x21	Multiple Authentications: Remotely Open Door
MINOR_MULTI_VERIFY_SUPERPASSWD_VERIFY_SUCCESS	0x22	Multiple Authentications: Super Password Authentication Completed
MINOR_MULTI_VERIFY_REPEAT_VERIFY	0x23	Multiple Authentications: Repeated Authentication
MINOR_MULTI_VERIFY_TIMEOUT	0x24	Multiple Authentications Timed Out
MINOR_DOORBELL_RINGING	0x25	Doorbell Ring
MINOR_FINGERPRINT_COMPARE_PASS	0x26	Fingerprint Matched
MINOR_FINGERPRINT_COMPARE_FAIL	0x27	Fingerprint Mismatched
MINOR_CARD_FINGERPRINT_VERIFY_PASS	0x28	Card and Fingerprint Authentication Completed
MINOR_CARD_FINGERPRINT_VERIFY_FAIL	0x29	Card and Fingerprint Authentication Failed
MINOR_CARD_FINGERPRINT_VERIFY_TIMEOUT	0x2a	Card and Fingerprint Authentication Timed Out
MINOR_CARD_FINGERPRINT_PASSWD_VERIFY_PASS	0x2b	Card and Fingerprint and Password Authentication Completed

Event Minor Types	Value	Description
MINOR_CARD_FINGERPRINT_PASSWD_VERIFY_FAIL	0x2c	Card and Fingerprint and Password Authentication Failed
MINOR_CARD_FINGERPRINT_PASSWD_VERIFY_TIMEOUT	0x2d	Card and Fingerprint and Password Authentication Timed Out
MINOR_FINGERPRINT_PASSWD_VERIFY_PASS	0x2e	Fingerprint and Password Authentication Completed
MINOR_FINGERPRINT_PASSWD_VERIFY_FAIL	0x2f	Fingerprint and Password Authentication Failed
MINOR_FINGERPRINT_PASSWD_VERIFY_TIMEOUT	0x30	Fingerprint and Password Authentication Timed Out
MINOR_FINGERPRINT_INEXISTENCE	0x31	Fingerprint Not Exists
MINOR_CARD_PLATFORM_VERIFY	0x32	Card Platform Authentication
MINOR_CALL_CENTER	0x33	Call Center
MINOR_FIRE_RELAY_TURN_ON_DOOR_ALWAYS_OPEN	0x34	Fire Relay Closed: Door Remains Open
MINOR_FIRE_RELAY_RECOVER_DOOR_RECOVER_NORMAL	0x35	Fire Relay Opened: Door Remains Closed
MINOR_EMPLOYEEENO_AND_FP_VERIFY_PASS	0x45	Employee ID and Fingerprint Authentication Completed
MINOR_EMPLOYEEENO_AND_FP_VERIFY_FAIL	0x46	Employee ID and Fingerprint Authentication Failed
MINOR_EMPLOYEEENO_AND_FP_VERIFY_TIMEOUT	0x47	Employee ID and Fingerprint Authentication Timed Out
MINOR_EMPLOYEEENO_AND_FP_AND_PW_VERIFY_PASS	0x48	Employee ID and Fingerprint and Password Authentication Completed
MINOR_EMPLOYEEENO_AND_FP_AND_PW_VERIFY_FAIL	0x49	Employee ID and Fingerprint and Password Authentication Failed

Event Minor Types	Value	Description
MINOR_EMPLOYEEENO_AND_FP_AND_PW_VERIFY_TIMEOUT	0x4a	Employee ID and Fingerprint and Password Authentication Timed Out
MINOR_FACE_VERIFY_PASS	0x4b	Face Authentication Completed
MINOR_FACE_VERIFY_FAIL	0x4c	Face Authentication Failed
MINOR_EMPLOYEEENO_AND_FACE_VERIFY_PASS	0x4d	Employee ID and Face Authentication Completed
MINOR_EMPLOYEEENO_AND_FACE_VERIFY_FAIL	0x4e	Employee ID and Face Authentication Failed
MINOR_EMPLOYEEENO_AND_FACE_VERIFY_TIMEOUT	0x4f	Employee ID and Face Authentication Timed Out
MINOR_FACE_RECOGNIZE_FAIL	0x50	Face Recognition Failed
MINOR_FIRSTCARD_AUTHORIZE_BEGIN	0x51	First Card Authorization Started
MINOR_FIRSTCARD_AUTHORIZE_END	0x52	First Card Authorization Ended
MINOR_DOORLOCK_INPUT_SHORT_CIRCUIT	0x53	Lock Input Short Circuit Attempts Alarm
MINOR_DOORLOCK_INPUT_BROKEN_CIRCUIT	0x54	Lock Input Open Circuit Attempts Alarm
MINOR_DOORLOCK_INPUT_EXCEPTION	0x55	Lock Input Exception Alarm
MINOR_DOORCONTACT_INPUT_SHORT_CIRCUIT	0x56	Contact Input Short Circuit Attempts Alarm
MINOR_DOORCONTACT_INPUT_BROKEN_CIRCUIT	0x57	Contact Input Open Circuit Attempts Alarm
MINOR_DOORCONTACT_INPUT_EXCEPTION	0x58	Contact Input Exception Alarm
MINOR_OPENBUTTON_INPUT_SHORT_CIRCUIT	0x59	Exit Button Input Short Circuit Attempts Alarm
MINOR_OPENBUTTON_INPUT_BROKEN_CIRCUIT	0x5a	Exit Button Input Open Circuit Attempts Alarm

Event Minor Types	Value	Description
MINOR_OPENBUTTON_INPUT_EXCEPTION	0x5b	Exit Button Input Exception Alarm
MINOR_DOORLOCK_OPEN_EXCEPTION	0x5c	Unlocking Exception
MINOR_DOORLOCK_OPEN_TIMEOUT	0x5d	Unlocking Timed Out
MINOR_FIRSTCARD_OPEN_WITHOUT_AUTHORIZE	0x5e	Unauthorized First Card Opening Failed
MINOR_CALL_LADDER_RELAY_BREAK	0x5f	Call Elevator Relay Open
MINOR_CALL_LADDER_RELAY_CLOSE	0x60	Call Elevator Relay Closed
MINOR_AUTO_KEY_RELAY_BREAK	0x61	Auto Button Relay Open
MINOR_AUTO_KEY_RELAY_CLOSE	0x62	Auto Button Relay Closed
MINOR_KEY_CONTROL_RELAY_BREAK	0x63	Button Relay Open
MINOR_KEY_CONTROL_RELAY_CLOSE	0x64	Button Relay Closed
MINOR_EMPLOYEEENO_AND_PW_PASS	0x65	Employee ID and Password Authentication Completed
MINOR_EMPLOYEEENO_AND_PW_FAIL	0x66	Employee ID and Password Authentication Failed
MINOR_EMPLOYEEENO_AND_PW_TIMEOUT	0x67	Employee ID and Password Authentication Timed Out
MINOR_HUMAN_DETECT_FAIL	0x68	Human Detection Failed
MINOR_PEOPLE_AND_ID_CARD_COMPARE_PASS	0x69	Person and ID Card Matched
MINOR_PEOPLE_AND_ID_CARD_COMPARE_FAIL	0x70	Person and ID Card Mismatched
MINOR_CERTIFICATE_BLOCKLIST	0x71	Blocklist Event

Event Minor Types	Value	Description
MINOR_LEGAL_MESSAGE	0x72	Valid Message
MINOR_ILLEGAL_MESSAGE	0x73	Invalid Message
MINOR_DOOR_OPEN_OR_DORMANT_FAIL	0x75	Authentication Failed: Door Remain Closed or Door in Sleeping Mode
MINOR_AUTH_PLAN_DORMANT_FAIL	0x76	Authentication Failed: Authentication Schedule in Sleeping Mode
MINOR_CARD_ENCRYPT_VERIFY_FAIL	0x77	Card Encryption Verification Failed
MINOR_SUBMARINEBACK_REPLY_FAIL	0x78	Anti-passing Back Server Response Failed
MINOR_DOOR_OPEN_OR_DORMANT_OPEN_FAIL	0x82	Open Door via Exit Button Failed When Door Remain Closed or in Sleeping Mode
MINOR_DOOR_OPEN_OR_DORMANT_LINKAGE_OPEN_FAIL	0x84	Door Linkage Open Failed During Door Remain Close or Sleeping
MINOR_TRAILING	0x85	Tailgating
MINOR_REVERSE_ACCESS	0x86	Reverse Passing
MINOR_FORCE_ACCESS	0x87	Force Accessing
MINOR_CLIMBING_OVER_GATE	0x88	Climb Over
MINOR_PASSING_TIMEOUT	0x89	Passing Timed Out
MINOR_INTRUSION_ALARM	0x8a	Intrusion Alarm
MINOR_FREE_GATE_PASS_NOT_AUTH	0x8b	Authentication Failed When Free Passing
MINOR_DROP_ARM_BLOCK	0x8c	Barrier Obstructed
MINOR_DROP_ARM_BLOCK_RESUME	0x8d	Barrier Restored
MINOR_PASSWORD_MISMATCH	0x97	Passwords Mismatched

Event Minor Types	Value	Description
MINOR_EMPLOYEE_NO_NOT_EXIST	0x98	Employee ID Not Exists
MINOR_COMBINED_VERIFY_PASS	0x99	Combined Authentication Completed
MINOR_COMBINED_VERIFY_TIMEOUT	0x9a	Combined Authentication Timed Out
MINOR_VERIFY_MODE_MISMATCH	0x9b	Authentication Type Mismatched
MINOR_BLUETOOTH_VERIFY_PASS	0x9f	Authenticated via Bluetooth
MINOR_BLUETOOTH_VERIFY_FAIL	0xa0	Authentication via Bluetooth Failed
MINOR_INFORMAL_MIFARE_CARD_VERIFY_FAIL	0xa2	Authentication Failed: Invalid Mifare Card
MINOR_CPU_CARD_ENCRYPT_VERIFY_FAIL	0xa3	Verifying CPU Card Encryption Failed
MINOR_NFC_DISABLE_VERIFY_FAIL	0xa4	Disabling NFC Verification Failed
MINOR_EM_CARD_RECOGNIZE_NOT_ENABLED	0xa8	EM Card Recognition Disabled
MINOR_M1_CARD_RECOGNIZE_NOT_ENABLED	0xa9	M1 Card Recognition Disabled
MINOR_CPU_CARD_RECOGNIZE_NOT_ENABLED	0xaa	CPU Card Recognition Disabled
MINOR_ID_CARD_RECOGNIZE_NOT_ENABLED	0xab	ID Card Recognition Disabled
MINOR_CARD_SET_SECRET_KEY_FAIL	0xac	Importing Key to Card Failed
MINOR_LOCAL_UPGRADE_FAIL	0xad	Local Upgrade Failed
MINOR_REMOTE_UPGRADE_FAIL	0xae	Remote Upgrade Failed
MINOR_REMOTE_EXTEND_MODULE_UPGRADE_SUCC	0xaf	Extension Module is Remotely Upgraded

Event Minor Types	Value	Description
MINOR_REMOTE_EXTEND_MODULE_UPGRADE_FAIL	0xb0	Upgrading Extension Module Remotely Failed
MINOR_REMOTE_FINGER_PRINT_MODULE_UPGRADE_SUCC	0xb1	Fingerprint Module is Remotely Upgraded
MINOR_REMOTE_FINGER_PRINT_MODULE_UPGRADE_FAIL	0xb2	Upgrading Fingerprint Module Remotely Failed
MINOR_DYNAMICCODE_VERIFY_PASS	0xb3	Dynamic Verification Code Authenticated
MINOR_DYNAMICCODE_VERIFY_FAIL	0xb4	Authentication with Verification Code Failed
MINOR_PASSWD_VERIFY_PASS	0xb5	Password Authenticated
MINOR_FULL_STAFF	0xc1	Number of People Exceeds 90% of Capacity
MINOR_BLUETOOTH_KEY_VERIFY_FAIL	/	Verifying Bluetooth Key Failed
MINOR_EVENT_CUSTOM1 to MINOR_EVENT_CUSTOM64	0x500 to 0x53f	Access Control: Custom Event 1 to Custom Event 64

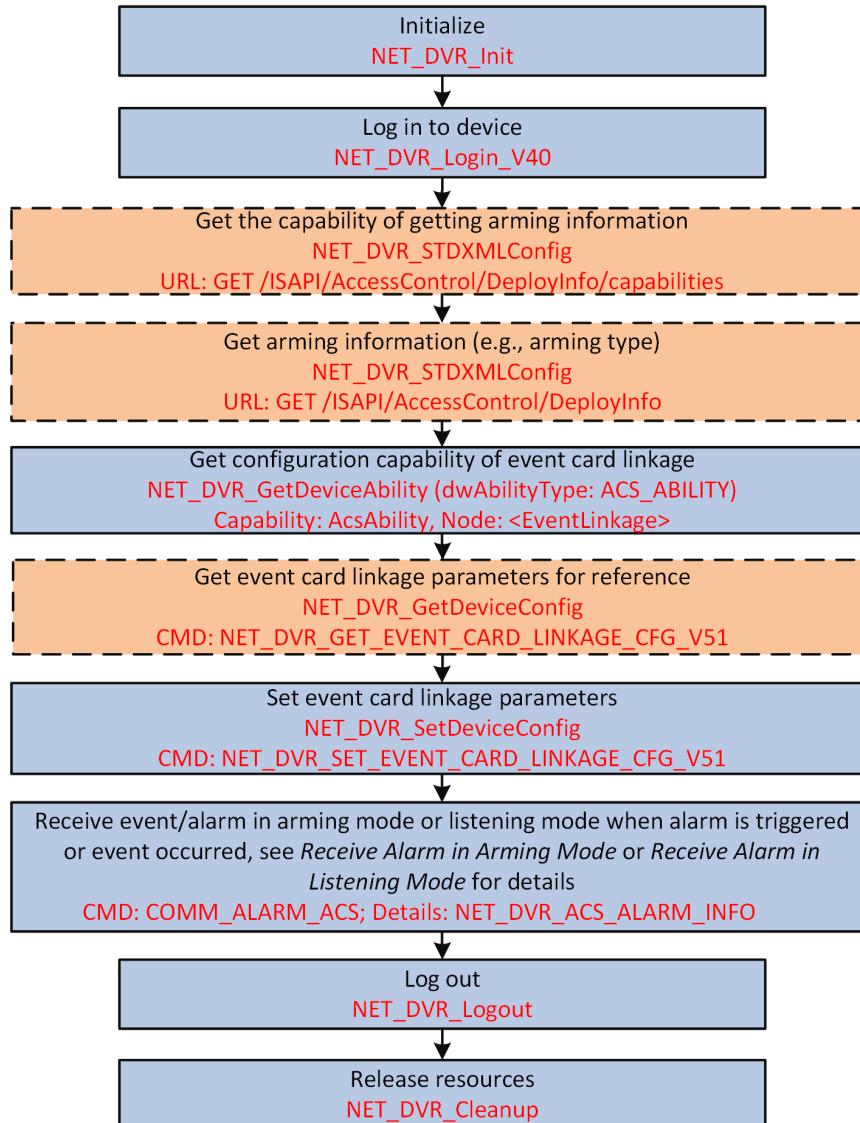
## 2.9.2 Configure Access Control Event

The access control events include device events, alarm input events, door events, card reader events, card swiping events, and so on. You can configure the linkage types (i.e., event linkage, card linkage, MAC linkage, and person linkage) and linkage actions (e.g., recording, alarm output, buzzing, capture, etc.) of event card linkage to execute the linked actions when the corresponding events occurred (e.g., door open or closed, card swiped, etc.). And then you can receive the event information from event sources in arming or listening mode.

### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to device.

## Steps



**Figure 1-18 Programming Flow of Configuring Access Control Event**

1. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [/ISAPI/AccessControl/DeployInfo/capabilities](#) for getting the capability of getting device arming information.  
The capability message [XML\\_Cap\\_DeployInfo](#) is returned.
2. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [/ISAPI/AccessControl/DeployInfo](#) for getting the device arming information to check whether the device is armed by other platforms or client software.
3. Call [NET\\_DVR\\_GetDeviceAbility](#) and set the capability type **dwAbilityType** to "ACS\_ABILITY" for getting the event card linkage configuration capability to know the configuration details or notices.



The input parameter pointer **pInBuf** should be set to the message [XML\\_Desc\\_AcsAbility](#).

---

The capability is returned in the message [XML\\_AcsAbility](#) by the output parameter pointer **pOutBuf**. The related nodes is [<EventLinkage>](#).

**4. Optional:** Call [NET\\_DVR\\_GetDeviceConfig](#) with

NET\_DVR\_GET\_EVENT\_CARD\_LINKAGE\_CFG\_V51 (command No.: 2518) to get the existing configurations for reference.

---



The parameter **dwCount** should be set to 1.

---

**5.** Call [NET\\_DVR\\_SetDeviceConfig](#) with NET\_DVR\_SET\_EVENT\_CARD\_LINKAGE\_CFG\_V51 (command No.: 2519) to set the event card linkage parameters.

---



- The parameter **lpInBuffer** refers to the structure [NET\\_DVR\\_EVENT\\_CARD\\_LINKAGE\\_COND](#), and the parameter **lpOutBuffer** or **lpInParamBuffer** refers to the structure [NET\\_DVR\\_EVENT\\_CARD\\_LINKAGE\\_CFG\\_V51](#).
- The parameter **dwCount** should be set to 1.

**6.** Receive event/alarm in arming mode (see [Receive Alarm/Event in Arming Mode](#)) or listening mode (see [Receive Alarm/Event in Listening Mode](#)) when alarm is triggered or event occurred.

---



The command to receive access control alarms/events should be set to COMM\_ALARMACS (command No.: 0x5002) in the APIs of [NET\\_DVR\\_SetDVRMessageCallBack\\_V50](#) and [NET\\_DVR\\_StartListen\\_V30](#), and refer to the data structure [NET\\_DVR\\_ACS\\_ALARM\\_INFO](#) for the alarm/event details.

---

**What to do next**

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out off the device and release the resources.

## 2.9.3 Supported Alarm/Event Types and Details

Event Type	ICommand (Command No.)	Event Details
Access Control Event	COMM_ALARMACS (0x5002)	<a href="#"><u>NET_DVR_ACS_ALARM_INFO</u></a>
ID Card Swiping Event	COMM_ID_INFO_ALARM (0x5200)	<a href="#"><u>NET_DVR_ID_CARD_INFO_ALARM</u></a>
QR Code Event	COMM_ISAPI_ALARM (0x6009)	<a href="#"><u>JSON_EventNotificationAlert_QRCodeEventMsg</u></a>

Event Type	ICommand (Command No.)	Event Details
		<p> <b>Note</b>            To check whether the device supports uploading QR code events, you can call <a href="#"><u>NET_DVR_STDXMLConfig</u></a> to transmit the request URL: <a href="#"><u>/ISAPI/System/capabilities</u></a> for getting the device capability. The device capability is returned in the message <a href="#"><u>XML_DeviceCap</u></a> by <b>IpOutputParam</b>. If uploading QR code events is supported, the node <b>&lt;isSupportQRCodeEvent&gt;</b> will be returned and its value is true.</p>
Face Temperature Screening Event		<p><a href="#"><u>JSON_EventNotificationAlert_FaceTempScreeningEventMsg</u></a></p> <p> <b>Note</b>            To check whether the device supports uploading face temperature screening events, you can call <a href="#"><u>NET_DVR_STDXMLConfig</u></a> to transmit the request URL: <a href="#"><u>/ISAPI/System/capabilities</u></a> for getting the device capability. The device capability is returned in the message <a href="#"><u>XML_DeviceCap</u></a> by <b>IpOutputParam</b>. If uploading face temperature screening events is supported, the node <b>&lt;isSupportFaceTemperatureMeasurement&gt;</b> will be returned and its value is true.</p>

## 2.9.4 Configure Mask Detection Event

You can configure mask detection parameters to determine whether to open the door or whether to prompt when the person does not wear a mask.

Function	Description
Get Configuration Capability of Mask Detection	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET <b><i>/ISAPI/AccessControl/maskDetection/capabilities?format=json</i></b> . The configuration capability is returned in <b><i>JSON_MaskDetectionCap</i></b> by <b>IpOutputParam</b> .
Get Mask Detection Parameters	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET <b><i>/ISAPI/AccessControl/maskDetection?format=json</i></b> . The parameters are returned in <b><i>JSON_MaskDetection</i></b> by <b>IpOutputParam</b> .
Set Mask Detection Parameters	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: PUT <b><i>/ISAPI/AccessControl/maskDetection?format=json</i></b> and set <b>IpInputParam</b> to <b><i>JSON_MaskDetection</i></b> .



To check whether the device supports mask detection, you can call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: GET ***/ISAPI/AccessControl/capabilities*** for getting the access control capability.

The access control capability is returned in the message ***XML\_Cap\_AccessControl*** by **IpOutputParam**. If the device supports mask detection, the node <isSupportMaskDetection> will be returned and its value is true.

## 2.9.5 Configure Hard Hat Detection Event

You can configure hard hat detection parameters to determine whether to open the door when the person does not wear a hard hat.

Function	Request URI
Get Configuration Capability of Hard Hat Detection	GET <b><i>/ISAPI/AccessControl/Configuration/safetyHelmetDetection/capabilities?format=json</i></b>
Get or Set Hard Hat Detection Parameters	GET or PUT <b><i>/ISAPI/AccessControl/Configuration/safetyHelmetDetection?format=json</i></b>



To check whether the device supports hard hat detection, you can call ***/ISAPI/AccessControl/capabilities*** by GET method to get the access control capability.

The access control capability is returned in the message `XML_Cap_AccessControl`. If the device supports hard hat detection, the node `<isSupportSafetyHelmetDetection>` will be returned and its value is true.

---

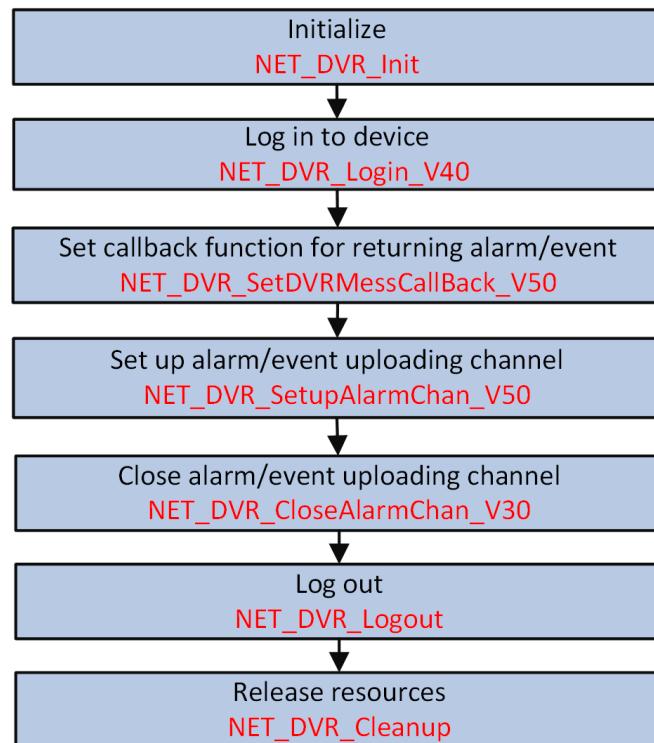
### 2.9.6 Receive Alarm/Event in Arming Mode

When the alarm is triggered or the event occurred, the secondarily developed third-party platform can automatically connect and send alarm/event uploading command to the device, and then the device uploads the alarm/event information to the platform for receiving.

#### Before You Start

- Make sure you have called `NET_DVR_Init` to initialize the development environment.
- Make sure you have called `NET_DVR_Login_V40` to log in to the device.
- Make sure you have configured the alarm/event parameters, refer to the typical alarm/event configurations for details.

#### Steps



**Figure 1-19 Programming Flow of Receiving Alarm/Event in Arming Mode**

1. Call `NET_DVR_SetDVRMessageCallBack_V50` to set callback function for returning alarm/event information.



## Note

- If the configured alarm is triggered or event occurred, the alarm/event information will be uploaded by device and returned in the callback function. You can view the alarm/event and do some processing operations.
- For the integration via device network SDK (HCNetSDK), to receive different types of alarm/event information, the parameter **ICommand** (data type to be uploaded) in the configured callback function should be different (refer to the typical alarm/event configurations). For the integration via text protocol, the **ICommand** should be set to "COMM\_ISAPI\_ALARM" (command No.: 0x6009) and the input parameter **pAlarmInfo** in the callback function **MSGCallBack** should be set to **NET\_DVR\_ALARM\_ISAPI\_INFO**.

2. Call **NET\_DVR\_SetupAlarmChan\_V50** to set up uploading channel.
3. Call **NET\_DVR\_CloseAlarmChan\_V30** to close uploading channel and stop receiving alarm or event information.

## Example

### Sample Code of Receiving Alarm or Event in Arming Mode

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

void main() {
    //-----
    // Initialize
    NET_DVR_Init();
    //Set connection time and reconnection time
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);
    //-----
    // Log in to device
    LONG lUserID;
    //Login parameters, including device IP address, user name, password, and so
    on.
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsynLogin = 0; //Synchronous login mode
    strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //Device IP address
    struLoginInfo.wPort = 8000; //Service port No.
    strcpy(struLoginInfo.sUserName, "admin"); //User name
    strcpy(struLoginInfo.sPassword, "abcd1234"); //Password
    //Device information, output parameter
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};
    lUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
    if (lUserID < 0)
    {
        printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }
```

```
}

//Set alarm callback function
NET_DVR_SetDVRMessageCallBack_V50(0, MessageCallbackNo1, NULL);
NET_DVR_SetDVRMessageCallBack_V50(1, MessageCallbackNo2, NULL);

//Enable arming
NET_DVR_SETUPALARMPARAM_V50 struSetupParamV50={0};
struSetupParamV50.dwSize=sizeof(NET_DVR_SETUPALARMPARAM_V50);
//Alarm category to be uploaded
struSetupParamV50.byAlarmInfoType=1;
//Arming level
struSetupParamV50.byLevel=1;

char szSubscribe[1024] = {0};
//The following code is for alarm subscription (subscribe all)
memcpy(szSubscribe, "<SubscribeEvent version=\"2.0\" xmlns=\"http://
www.isapi.org/ver20/XMLSchema\">>\r\n<eventMode>all</eventMode>\r\n", 1024);
LONG lHandle = -1;
if (0 == strlen(szSubscribe))
{
    //Arm
    lHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50, NULL,
strlen(szSubscribe));
}
else
{
    //Subscribe
    lHandle = NET_DVR_SetupAlarmChan_V50(lUserID, &struSetupParamV50,
szSubscribe, strlen(szSubscribe));
}

if (lHandle < 0)
{
    printf("NET_DVR_SetupAlarmChan_V50 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(20000);
//Disarm the uploading channel
if (!NET_DVR_CloseAlarmChan_V30(lHandle))
{
    printf("NET_DVR_CloseAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Log out
NET_DVR_Logout(lUserID);
```

```
//Release resources  
NET_DVR_Cleanup();  
return;  
}
```

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release resources.

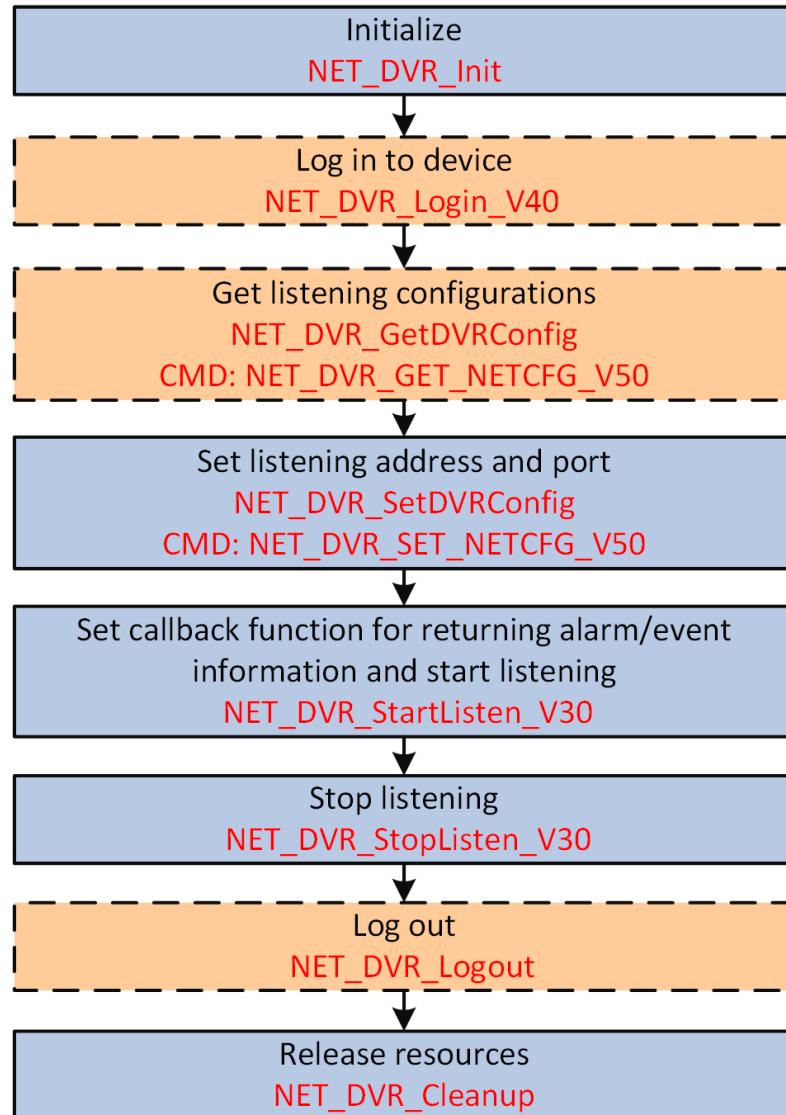
### 2.9.7 Receive Alarm/Event in Listening Mode

When alarm is triggered or event occurred, the device uploads the alarm/event information automatically, so you can configure the listening address and port for listening and receiving the alarm/event in the secondarily developed third-part platform.

#### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have configured the alarm/event parameters, refer to the typical alarm/event configurations for details.

## Steps



**Figure 1-20 Programming Flow of Receiving Alarm/Event in Listening Mode**

1. **Optional:** Call [NET\\_DVR\\_Login\\_V40](#) to log in to device.
2. **Optional:** Call [NET\\_DVR\\_GetDVRConfig](#) with "NET\_DVR\_GET\_NETCFG\_V50" (command No.: 1015) to get the existing listening configurations (i.e., listening address and port) for reference. The listening parameters are retruned in the structure [NET\\_DVR\\_NETCFG\\_V50](#) by the output parameter pointer **lpOutBuffer**.
3. Call [NET\\_DVR\\_SetDVRConfig](#) with "NET\_DVR\_SET\_NETCFG\_V50" (command No.: 1016) and specify the input parameter pointer **lpInBuffer** to the structure [NET\\_DVR\\_NETCFG\\_V50](#) for setting the listening address and port.
4. Call [NET\\_DVR\\_StartListen\\_V30](#) to set callback function for returning alarm/event information and start the listening.



## Note

For the integration via device network SDK (HCNetSDK), to receive different types of alarm/event information, the parameter **ICommand** (data type to be uploaded) in the configured callback function should be different (refer to the typical alarm/event configurations). For the integration via text protocol, the **ICommand** should be set to "COMM\_ISAPI\_ALARM" and the input parameter **pAlarmInfo** in the callback function **MSGCallBack** should be set to **NET\_DVR\_ALARM\_ISAPI\_INFO**.

---

The alarm/event information is automatically uploaded by the device when the configured alarm is triggered or event occurred, and the third-party platform or system gets the alarm/event information from the configured callback function.

5. Call **NET\_DVR\_StopListen\_V30** to stop listening and receiving alarm or event information.

### Example

#### Sample Code of Receiving Alarm/Event in Listening Mode

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;
void main() {
    //-----
    // Initialize
    NET_DVR_Init();
    //Set connection time and reconnection time
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);
    //-----
    // Log in to device
    LONG lUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    lUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345",
&struDeviceInfo);
    if (lUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }
    //Enable listening
    LONG lHandle;
    lHandle = NET_DVR_StartListen_V30(NULL, 7200, MessageCallback, NULL);
    if (lHandle < 0)
    {
        printf("NET_DVR_StartListen_V30 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(lUserID);
        NET_DVR_Cleanup();
        return;
    }
}
```

```
    Sleep(5000);
    //Disable listening
    if (!NET_DVR_StopListen_V30(lHandle))
    {
        printf("NET_DVR_StopListen_V30 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(lUserID);
        NET_DVR_Cleanup();
        return;
    }
    //Log out
    NET_DVR_Logout(lUserID);
    //Release SDK resource
    NET_DVR_Cleanup();
    return;
}
```

## What to do next

Call [NET\\_DVR\\_Logout](#) (if logged in) and [NET\\_DVR\\_Cleanup](#) to log out and release resources.

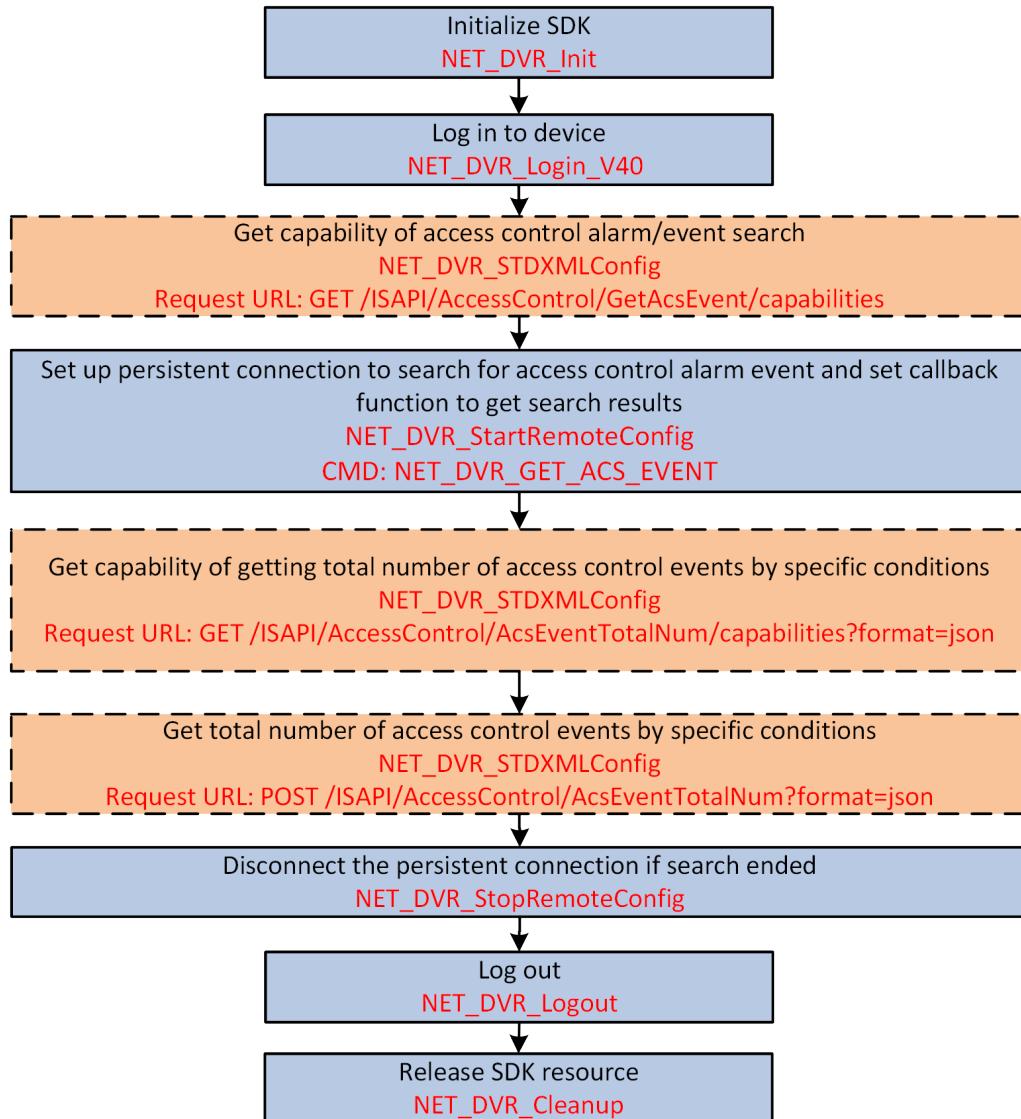
### 2.9.8 Search for Access Control Events

If the access control alarms or events are received and stored in the third-party platform, you can search for the alarms or events by setting different search conditions.

#### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to device.

## Steps



**Figure 1-21 Programming Flow of Searching for Access Control Events**

1. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [GET /ISAPI/AccessControl/GetAcsEvent/capabilities](#) for getting the capability of access control alarm/event search to know the details or notices about search.  
The capability message [XML\\_Cap\\_GetAcsEvent](#) is returned.
2. Call [NET\\_DVR\\_StartRemoteConfig](#) with  
[NET\\_DVR\\_GET\\_ACS\\_EVENT](#)  
(command No: 2514) and set input buffer ([IpInBuffer](#)) to [NET\\_DVR\\_ACS\\_EVENT\\_COND](#) for setting up persistent connection and set callback function ([fRemoteConfigCallback](#)).  
The access control event details are returned in the structure [NET\\_DVR\\_ACS\\_EVENT\\_CFG](#) by the output buffer ([IpBuffer](#)) of callback function.

- 3. Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET /ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json to get the capability of getting total number of access control events by specific conditions.
- 4. Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: POST /ISAPI/AccessControl/AcsEventTotalNum?format=json and set input buffer (IplnBuffer) to the message JSON\_AcsEventTotalNumCond for getting the total number of access control events by specific conditions.
- 5.** Call [NET\\_DVR\\_StopRemoteConfig](#) to disconnect the persistent connection and finish searching.

### Example

#### Sample Code of Searching for Access Control Event

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

BOOL CALLBACK MSesGCallback(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char
*pAlarmInfo, DWORD dwBufLen, void* pUser)
{
    //As the operations with long time consumption are not allowed in the
    //callback function,
    //do not call the API of HCNetSDK.DLL in the callback function.
    //The following code is for reference only, actually, processing data in
    //the callback function is not suggested.
    //for example, process in the message response function as PostMessage
    switch (lCommand)
    {
        case COMM_ALARM_ACN://Alarm information of access controller
        {
            NET_DVR_ACN_ALARM_INFO struAcsAlarmInfo = {0};
            memcpy(&struAcsAlarmInfo, pAlarmInfo, sizeof(struAcsAlarmInfo));
            //Handle other information in the alarm structure as desired...
            break;
        }
        case COMM_PASSTIME_INFO_ALARM://Number of passed persons
        {
            NET_DVR_PASSTIME_INFO_ALARM struPassnumInfo = {0};
            memcpy(&struPassnumInfo, pAlarmInfo, sizeof(struPassnumInfo));
            //Handle other information in the alarm structure as desired...
            break;
        }
        default:
            break;
    }
    return true;
}
void main()
{
    //-----
}
```

```

//Initialize
NET_DVR_Init();

//Set connection timeout and reconnection function
NET_DVR_SetConnectTime(2000, 1);
NET_DVR_SetReconnect(10000, true);
//-----
//Log in to device
LONG lUserID;
//Login parameters, including device IP address, user name, password, and
so on
NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
struLoginInfo.bUseAsynLogin = 0; //Synchronous login mode
strcpy(struLoginInfo.sDeviceAddress, "192.168.1.64"); //Device IP address
struLoginInfo.wPort = 8000; //Device service port number
strcpy(struLoginInfo.sUserName, "admin"); //User name
strcpy(struLoginInfo.sPassword, "abcd1234"); //Password

//Device information, output parameter
NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

lUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
if (lUserID < 0)
{
    printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//Set alarm callback function for card swiping event
NET_DVR_SetDVRMessageCallBack_V31(MSesGCallback, NULL);
//Set up channel for uploading alarm information
NET_DVR_SETUPALARM_PARAM struSetupParam={0};
struSetupParam.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM);

LONG lHandle = NET_DVR_SetupAlarmChan_V41(lUserID,&struSetupParam);
if (lHandle < 0)
{
    printf("NET_DVR_SetupAlarmChan_V41 error, %d\n",
NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//Wait for 60s for receiving captured picture uploaded by device
Sleep(60000);
//Close alarm uploading channel
if (!NET_DVR_CloseAlarmChan_V30(lHandle))
{
    printf("NET_DVR_CloseAlarmChan_V30 error, %d\n",
NET_DVR_GetLastError());
}

```

```

        NET_DVR_Logout(lUserID);
        NET_DVR_Cleanup();
        return;
    }
    //Log out
    NET_DVR_Logout(lUserID);
    //Release SDK resource
    NET_DVR_Cleanup();
    return;
}

```

## What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out and release the resource.

## 2.9.9 Remotely Verify Access Control Events

For the uploaded access control events, you can verify them to control opening or closing the door.

Function	Description
Get Capability of Verifying Access Control Event Remotely	Call <a href="#"><u>NET_DVR_STDXMLConfig</u></a> to transmit the request URI: GET <a href="#"><u>/ISAPI/AccessControl/remoteCheck/capabilities?format=json</u></a> . The capability is returned in <a href="#"><u>JSON_Cap_RemoteCheck</u></a> by <a href="#"><u>IpOutputParam</u></a> .
Verify Access Control Event Remotely	Call <a href="#"><u>NET_DVR_STDXMLConfig</u></a> to transmit the request URI: PUT <a href="#"><u>/ISAPI/AccessControl/remoteCheck?format=json</u></a> and set <a href="#"><u>IpInputParam</u></a> to <a href="#"><u>JSON_RemoteCheck</u></a> .



To check whether the device supports verifying access control events remotely, you can call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/capabilities](#) for getting the access control capability.

The access control capability is returned in the message [XML\\_Cap\\_AccessControl](#) by [IpOutputParam](#). If the device supports this function, the node <isSupportRemoteCheck> will be returned and its value is true.

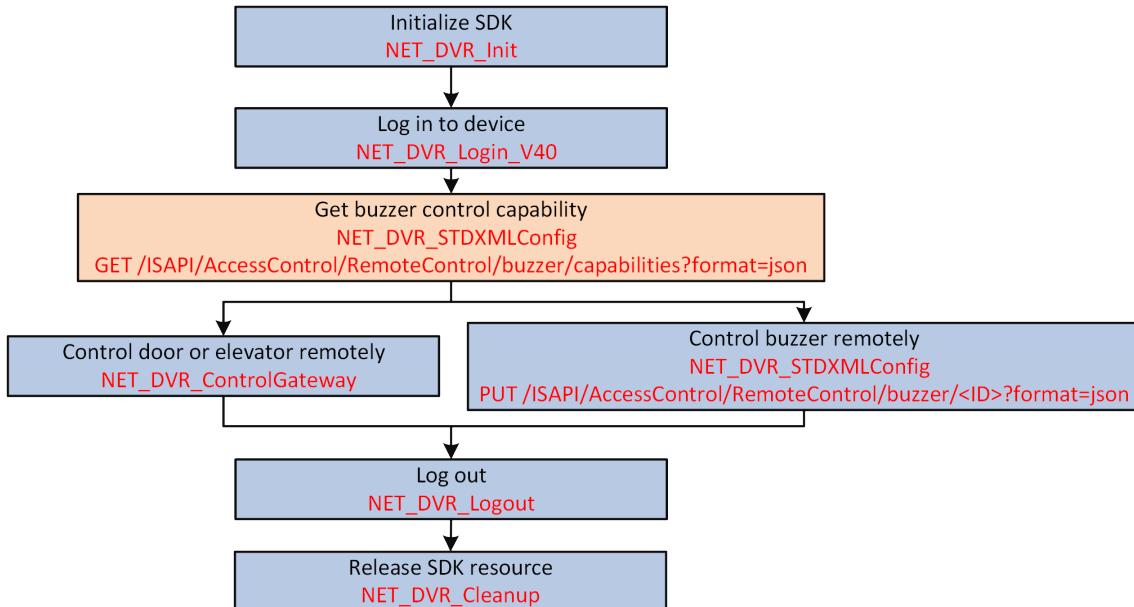
## 2.10 Remotely Control Door, Elevator, and Buzzer

You can remotely control the status of doors or elevators, and buzzer (i.e., start or stop buzzing).

### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the development environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to device.

## Steps



**Figure 1-22 Programming Flow of Remotely Control Door, Elevator, and Buzzer**

1. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [GET /ISAPI/AccessControl/RemoteControl/buzzer/capabilities?format=json](#) for getting the capability to judge whether the buzz control is supported and know the control details or notices.



The obtained capability ([JSON Cap\\_RemoteControlBuzzer](#)) is for buzz control only, the door control capability does not exist.

2. Perform one of the following operations to control door or elevator remotely or control buzz remotely.
  - Call [NET\\_DVR\\_ControlGateway](#) to control door or elevator remotely (i.e., open door, close door, remain open, remain closed).
  - Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: [PUT /ISAPI/AccessControl/RemoteControl/buzzer/<ID>?format=json](#) for controlling the buzz (i.e., start or stop buzzing).

### What to do next

Call [NET\\_DVR\\_Logout](#) and [NET\\_DVR\\_Cleanup](#) to log out off the device and release the resources.

## 2.11 Configure Attendance Status and Schedule

The time and attendance refers to tracking and monitoring when employees start and stop working, and their working hours (including late arrivals, early departures, time taken on breaks and absenteeism, etc.). You can set the manual or automatic time and attendance mode, or disable

the attendance mode. You can also configure the week schedule to regularly manage and control the attendance (i.e., check in, check out, break out, break in, overtime in, or overtime out) in some specific time periods.

### Before You Start

- Make sure you have called [\*\*NET\\_DVR\\_Init\*\*](#) to initialize the development environment.
- Make sure you have called [\*\*NET\\_DVR\\_Login\\_V40\*\*](#) to log in to the device.
- Make sure you have added at least one person, refer to [\*\*Manage Person Information\*\*](#) for details.

## Steps



**Figure 1-23 API Calling Flow of Configuring Attendance Status and Schedule**

1. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/Configuration/attendanceMode/capabilities?format=json](#) for getting the configuration capability of the attendance mode and knowing the configuration details and notices.



To check whether the device supports configuring the attendance mode, you can call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/capabilities](#) for getting the functional capability of access control.

If the node <isSupportAttendanceMode> is returned in the message [XML\\_Cap\\_AccessControl](#) by **IpOutputParam** and its value is true, it indicates that the device supports configuring the attendance mode.

---

The configuration capability is returned in the message [JSON\\_Cap\\_AttendanceMode](#) by **IpOutputParam**.

2. Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: PUT [/ISAPI/AccessControl/Configuration/attendanceMode?format=json](#) and set **IpInputParam** to the message [JSON\\_AttendanceMode](#) for setting the attendance mode parameters.



Before setting the attendance mode parameters, you'd better call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/Configuration/attendanceMode?format=json](#) for getting the existing or default attendance mode parameters for reference. The parameters are returned in the message [JSON\\_AttendanceMode](#) by **IpOutputParam**.

3. **Optional:** Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/keyCfg/attendance/capabilities?format=json](#) for getting the configuration capability of attendance check by pressing the key and knowing the configuration details and notices.



To check whether the device supports configuring parameters of attendance check by pressing the key, you can call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/capabilities](#) for getting the functional capability of access control.

If the node <isSupportKeyCfgAttendance> is returned in the message [XML\\_Cap\\_AccessControl](#) by **IpOutputParam** and its value is true, it indicates that the device supports configuring parameters of attendance check by pressing the key.

---

The configuration capability is returned in the message [JSON\\_AttendanceCap](#) by **IpOutputParam**.

4. Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: PUT [/ISAPI/AccessControl/keyCfg/<ID>/attendance?format=json](#) and set **IpInputParam** to the message [JSON\\_Attendance](#) for setting the parameters of attendance check by pressing the key.



Before setting the parameters, you'd better call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/keyCfg/<ID>/attendance?format=json](#) or [/ISAPI/](#)

**AccessControl/keyCfg/attendance?format=json** for getting the existing or default parameters of one or all keys for reference. The parameters are returned in the message **JSON\_Attendance** or **JSON\_AttendanceList** by **IpOutputParam**.

5. **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/Attendance/weekPlan/capabilities?format=json** for getting the configuration capability of the week attendance schedule and knowing the configuration details and notices.
- 



### Note

To check whether the device supports configuring the week attendance schedule, you can call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/capabilities** for getting the functional capability of access control.

If the node **<isSupportAttendanceWeekPlan>** is returned in the message **XML\_Cap\_AccessControl** by **IpOutputParam** and its value is true, it indicates that the device supports configuring the week attendance schedule.

---

The configuration capability is returned in the message **JSON\_AttendanceWeekPlanCap** by **IpOutputParam**.

6. Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json** and set **IpInputParam** to the message **JSON\_AttendanceWeekPlan** for setting the parameters of the week attendance schedule.
- 



Before setting the parameters, you'd better call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json** for getting the existing or default parameters for reference. The parameters are returned in the message **JSON\_AttendanceWeekPlan** by **IpOutputParam**.

7. **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/Attendance/planTemplate/capabilities?format=json** for getting the configuration capability of the attendance schedule template and knowing the configuration details and notices.
- 



To check whether the device supports configuring the attendance schedule template, you can call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/capabilities** for getting the functional capability of access control.

If the node **<isSupportAttendancePlanTemplate>** is returned in the message **XML\_Cap\_AccessControl** by **IpOutputParam** and its value is true, it indicates that the device supports configuring the attendance schedule template.

---

The configuration capability is returned in the message **JSON\_AttendancePlanTemplateCap** by **IpOutputParam**.

8. **Optional:** Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/Attendance/planTemplate?format=json** for getting the list of attendance schedule templates.
-



To check whether the device supports getting the list of attendance schedule templates, you can call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/capabilities** for getting the functional capability of access control.

If the node <isSupportAttendancePlanTemplateList> is returned in the message **XML\_Cap\_AccessControl** by **IpOutputParam** and its value is true, it indicates that the device supports getting the list of attendance schedule templates.

---

The list is returned in the message **JSON\_AttendancePlanTemplateList** by **IpOutputParam**.

9. Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json** and set **IpInputParam** to the message **JSON\_AttendancePlanTemplate** for setting the parameters of the attendance schedule template.
- 



Before setting the parameters, you'd better call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json** for getting the existing or default parameters for reference. The parameters are returned in the message **JSON\_AttendancePlanTemplate** by **IpOutputParam**.

10. Optional: Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/ClearAttendancePlan?format=json** and set **IpInputParam** to the message **JSON\_ClearAttendancePlan** for clearing the attendance schedule.
- 



To check whether the device supports clearing the attendance schedule, you can call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/capabilities** for getting the functional capability of access control.

If the node <isSupportClearAttendancePlan> is returned in the message **XML\_Cap\_AccessControl** by **IpOutputParam** and its value is true, it indicates that the device supports clearing the attendance schedule.

---

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out of the device and release the resources.

## 2.12 Turnstile Settings

The turnstile is a lane management device that is used to manage the entrance and exit of people in places such as office buildings, subways, residences, and so on. By adopting the turnstile integrated with the access control system, people should authenticate to pass through the lane by swiping ID card, scanning QR code, etc. Common turnstiles include swing barrier, flap barrier, tripod turnstile, and so on.

## Lane Controller Settings

The lane controller is mainly used to control infrared or motor components of the turnstile.

Function	Description
Get Configuration Capability of Lane Controller	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET / <b><i>ISAPI/AccessControl/ChannelControllerCfg/capabilities</i></b> . The configuration capability is returned in the message <b><i>XML_Cap_ChannelControllerCfg</i></b> by <b>IpOutputParam</b> .
Get Lane Controller Parameters	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET / <b><i>ISAPI/AccessControl/ChannelControllerCfg</i></b> . The parameters are returned in the message <b><i>XML_ChannelControllerCfg</i></b> by <b>IpOutputParam</b> .
Set Lane Controller Parameters	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: PUT / <b><i>ISAPI/AccessControl/ChannelControllerCfg</i></b> and set <b>IpInputParam</b> to the message <b><i>XML_ChannelControllerCfg</i></b> .



To check whether configuring lane controller parameters is supported, you can call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: GET /***ISAPI/AccessControl/capabilities*** to get the access control capability.

The access control capability is returned in the message ***XML\_Cap\_AccessControl*** by **IpOutputParam**. If configuring lane controller parameters is supported, the node <isSupportChannelControllerCfg> will be returned and its value is "true".

## Device Type Settings of Lane Controller

Function	Description
Get Configuration Capability of Device Type of Lane Controller	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET / <b><i>ISAPI/AccessControl/channelControllerTypeCfg/capabilities?format=json</i></b> . The configuration capability is returned in the message <b><i>JSON_ChannelControllerTypeCfgCap</i></b> by <b>IpOutputParam</b> .
Get Device Type Parameters of Lane Controller	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET / <b><i>ISAPI/AccessControl/channelControllerTypeCfg?format=json</i></b> .

Function	Description
	The device type parameters are returned in the message <b><i>JSON_ChannelControllerTypeCfg</i></b> by <b>IpOutputParam</b> .
Set Device Type Parameters of Lane Controller	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: PUT <b>/ISAPI/AccessControl/channelControllerTypeCfg?format=json</b> and set <b>IpInputParam</b> to the message <b><i>JSON_ChannelControllerTypeCfg</i></b> .



To check whether configuring device type parameters of the lane controller is supported, you can call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: GET **/ISAPI/AccessControl/capabilities** to get the access control capability.

The access control capability is returned in the message ***XML\_Cap\_AccessControl*** by **IpOutputParam**. If configuring device type parameters of the lane controller is supported, the node <isSupportChannelControllerTypeCfg> will be returned and its value is "true".

## Keyfob Control Mode

Function	Description
Get configuration capability of keyfob control mode	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET <b>/ISAPI/AccessControl/remoteCtrlModeCfg/capabilities?format=json</b> . The capability is returned in the message <b><i>JSON_RemoteCtrlModeCfgCap</i></b> by <b>IpOutputParam</b> .
Get parameters of keyfob control mode	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: GET <b>/ISAPI/AccessControl/remoteCtrlModeCfg?format=json</b> . The parameters are returned in the message <b><i>JSON_RemoteCtrlModeCfg</i></b> by <b>IpOutputParam</b> .
Set parameters of keyfob control mode	Call <b><i>NET_DVR_STDXMLConfig</i></b> to transmit the request URI: PUT <b>/ISAPI/AccessControl/remoteCtrlModeCfg?format=json</b> and set <b>IpInputParam</b> to the message <b><i>JSON_RemoteCtrlModeCfg</i></b> .



To check whether the device supports configuring parameters of the keyfob control mode, you can call ***NET\_DVR\_STDXMLConfig*** to transmit the request URI: GET **/ISAPI/AccessControl/capabilities** to get the access control capability.

The capability is returned in the message [XML\\_Cap\\_AccessControl](#) by **IpOutputParam**. If this function is supported by the device, the node <isSupportRemoteCtrlleModeCfg> will be returned in the message and its value is "true".

---

## 2.13 Other Applications

### 2.13.1 Device Settings

#### Door/Floor

##### Get door (floor) parameters

Call [NET\\_DVR\\_GetDVRConfig](#) with the command "NET\_DVR\_GET\_DOOR\_CFG" (command No.: 2108) and set **IChannel** to the door (floor) No. (it starts from 1).

And the configuration parameters are returned in the structure [NET\\_DVR\\_DOOR\\_CFG](#) by the output buffer (**IpOutBuffer**).

##### Set door (floor) parameters

Call [NET\\_DVR\\_SetDVRConfig](#) with the command "NET\_DVR\_SET\_DOOR\_CFG" (command No.: 2109), set **IChannel** to the door (floor) No. (it starts from 1), and set the input buffer (**IpInBuffer**) to the structure [NET\\_DVR\\_DOOR\\_CFG](#).



#### Note

To check whether the device supports door parameter configuration, you can call [NET\\_DVR\\_GetDeviceAbility](#), set the capability type **dwAbilityType** to "ACS\_ABILITY", and set the input parameter pointer **pInBuf** to the message [XML\\_Desc\\_AcsAbility](#) for getting the access control capability.

The capability is returned in the message [XML\\_AcsAbility](#) by the output parameter pointer **pOutBuf**. The related node is <Door>.

---

#### Fingerprint and Card Reader

##### Get fingerprint and card reader parameters

Call [NET\\_DVR\\_GetDVRConfig](#) with the command of NET\_DVR\_GET\_CARD\_READER\_CFG\_V50 (command No.: 2505).

And the configuration parameters are returned in the structure

[NET\\_DVR\\_CARD\\_READER\\_CFG\\_V50](#) by the output buffer (**IpOutBuffer**).

##### Set fingerprint and card reader parameters

Call [NET\\_DVR\\_SetDVRConfig](#) with the command of NET\_DVR\_SET\_CARD\_READER\_CFG\_V50 (command No.: 2506) and set the input buffer (**IpInBuffer**) to the structure

[NET\\_DVR\\_CARD\\_READER\\_CFG\\_V50](#).

## NFC (Near-Field Communication) Function

### Get configuration capability of enabling or disabling NFC function

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/Configuration/NFCCfg/capabilities?format=json](#).

And the configuration capability is returned in the message [JSON\\_NFCCfgCap](#) by the output parameter (IpOutputParam).

### Get parameters of enabling or disabling NFC function

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/Configuration/NFCCfg?format=json](#).

And the parameters are returned in the message [JSON\\_NFCCfg](#) by IpOutBuffer of IpOutputParam.

### Set parameters of enabling or disabling NFC function

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/Configuration/NFCCfg?format=json](#) and set IpInBuffer of IpInputParam to the message [JSON\\_NFCCfg](#).

## RF (Radio Frequency) Card Recognition

### Get configuration capability of enabling or disabling RF card recognition

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/Configuration/RFCardCfg/capabilities?format=json](#).

And the configuration capability is returned in the message [JSON\\_RFCardCfgCap](#) by the output parameter (IpOutputParam).

### Get parameters of enabling or disabling RF card recognition

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/Configuration/RFCardCfg?format=json](#).

And the parameters are returned in the message [JSON\\_RFCardCfg](#) by IpOutBuffer of IpOutputParam.

### Set parameters of enabling or disabling RF card recognition

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/Configuration/RFCardCfg?format=json](#) and set IpInBuffer of IpInputParam to the message [JSON\\_RFCardCfg](#).

## Access Controller

### Get configuration capability of access controller

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/AcsCfg/capabilities?format=json](#).

The configuration capability is returned in [JSON\\_Cap\\_AcsCfg](#) by IpOutputParam.

### Get access controller parameters

- Call **NET\_DVR\_GetDVRConfig** with the command of NET\_DVR\_GET\_ACS\_CFG (command No.: 2159).  
And the configuration parameters are returned in the structure **NET\_DVR\_ACS\_CFG** by the output buffer (**IpOutBuffer**).
- Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/AcsCfg?format=json**.  
The parameters are returned in **JSON\_AcsCfg** by **IpOutputParam**.

### Set access controller parameters

- Call **NET\_DVR\_SetDVRConfig** with the command of NET\_DVR\_SET\_ACS\_CFG (command No.: 2160) and set the input buffer (**IpInBuffer**) to the structure **NET\_DVR\_ACS\_CFG**.
- Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/AcsCfg?format=json** and set **IpInputParam** to **JSON\_AcsCfg**.

## OSDP (Open Supervised Device Protocol) Card Reader

### Get capability of getting the OSDP card reader status

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/OSDPStatus/capabilities?format=json**.

And the capability is returned in the message **JSON\_Cap\_OSDPStatus** by the output parameter (**IpOutputParam**).

### Get OSDP card reader status

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/OSDPStatus/<ID>?format=json**.

And the parameters are returned in the message **JSON\_OSDPStatus** by the output buffer (**IpOutBuffer**) of the output parameter (**IpOutputParam**).

### Get capability of setting OSDP card reader ID

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/OSDPMODify/capabilities?format=json**.

And the configuration capability is returned in the message **JSON\_Cap\_OSDPMODify** by the output parameter (**IpOutputParam**).

### Set OSDP card reader ID

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: PUT **/ISAPI/AccessControl/OSDPMODify/<ID>?format=json** and set the input buffer (**IpInBuffer**) of the input parameter (**IpInputParam**) to the message **JSON\_OSDPMODify**.

## Intelligent Identity Recognition Terminal

### Get configuration capability of intelligent identity recognition terminal

Call **NET\_DVR\_STDXMLConfig** to pass through the request URL: GET **/ISAPI/AccessControl/IdentityTerminal/capabilities**.

And the configuration capability is returned in the message [XML\\_Cap\\_IdentityTerminal](#) by the output parameter (**IpOutputParam**).

### Get parameters of intelligent identity recognition terminal

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/AccessControl/IdentityTerminal](#).

And the parameters are returned in the message [XML\\_IdentityTerminal](#) by the output buffer (**IpOutBuffer**) of the output parameter (**IpOutputParam**).

### Set parameters of intelligent identity recognition terminal

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/AccessControl/IdentityTerminal](#) and set the input buffer (**IpInBuffer**) of the input parameter (**IpInputParam**) to the message [XML\\_IdentityTerminal](#).

## Picture Storage Server

### Get picture storage server parameters

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: GET [/ISAPI/System/PictureServer?format=json](#).

And the parameters are returned in the message [JSON\\_PictureServerInformation](#) by the output parameter **IpOutputParam**.

### Set picture storage server parameters

Call [NET\\_DVR\\_STDXMLConfig](#) to pass through the request URL: PUT [/ISAPI/System/PictureServer?format=json](#) and set the input parameter **IpInputParam** to the message [JSON\\_PictureServerInformation](#).

## 2.13.2 Multi-Factor Authentication

The multi-factor authentication is to manage the cards by group and set the authentication for multiple cards of one access control point (door).

### Parameter Settings

- Get multi-factor authentication parameters

Call [NET\\_DVR\\_GetDVRConfig](#) with the command of NET\_DVR\_GET\_MULTI\_CARD\_CFG\_V50 (command No.: 2515).

And the configuration parameters are returned in the structure

[NET\\_DVR\\_MULTI\\_CARD\\_CFG\\_V50](#) by **IpOutBuffer**.

- Set multi-factor authentication parameters

Call [NET\\_DVR\\_SetDVRConfig](#) with the command of NET\_DVR\_SET\_MULTI\_CARD\_CFG\_V50 (command No.: 2516) and set **IpInBuffer** to the structure [NET\\_DVR\\_MULTI\\_CARD\\_CFG\\_V50](#).

## Group Settings

- Get group parameters  
Call **NET\_DVR\_GetDVRConfig** with the command of NET\_DVR\_GET\_GROUP\_CFG (command No.: 2112).  
And the configuration parameters are returned in the structure **NET\_DVR\_GROUP\_CFG** by **IpOutBuffer**.
- Set group parameters  
Call **NET\_DVR\_SetDVRConfig** with the command of NET\_DVR\_SET\_GROUP\_CFG (command No.: 2113) and set **IpInBuffer** to the structure **NET\_DVR\_GROUP\_CFG**.

### 2.13.3 Temperature Measurement



#### Note

For integration via Device Network SDK, the related text protocol data should be transmitted by the specific API (i.e., **NET\_DVR\_STDXMLConfig**) to realize the applications. Refer to **Integrate by Transmitting Text Protocol** for details.

#### Temperature Measurement Area

Function	Request URI
Get the configuration capability of the temperature measurement area	GET <b><u>/ISAPI/AccessControl/temperatureMeasureAreaCfg/capabilities?format=json</u></b>
Get the parameters of the temperature measurement area	GET <b><u>/ISAPI/AccessControl/temperatureMeasureAreaCfg?format=json</u></b>
Set the parameters of the temperature measurement area	PUT <b><u>/ISAPI/AccessControl/temperatureMeasureAreaCfg?format=json</u></b>

#### Temperature Measurement Area Calibration

Function	Request URI
Get the calibration configuration capability of the	GET <b><u>/ISAPI/AccessControl/temperatureMeasureAreaCalibration/capabilities?format=json</u></b>

Function	Request URI
temperature measurement area	
Get the calibration parameters of the temperature measurement area	GET <a href="#"><u>/ISAPI/AccessControl/temperatureMeasureAreaCalibration?format=json</u></a>
Set the calibration parameters of the temperature measurement area	PUT <a href="#"><u>/ISAPI/AccessControl/temperatureMeasureAreaCalibration?format=json</u></a>

## Temperature Measurement Settings

Function	Request URI
Get the configuration capability of temperature measurement parameters	GET <a href="#"><u>/ISAPI/AccessControl/temperatureMeasureCfg/capabilities?format=json</u></a>
Get temperature measurement parameters	GET <a href="#"><u>/ISAPI/AccessControl/temperatureMeasureCfg?format=json</u></a>
Set temperature measurement parameters	PUT <a href="#"><u>/ISAPI/AccessControl/temperatureMeasureCfg?format=json</u></a>

### 2.13.4 Other Configurations

#### Working Status

Call [NET\\_DVR\\_GetDVRConfig](#) with the command of "NET\_DVR\_GET\_ACS\_WORK\_STATUS\_V50" (command No.: 2180).

And the working status is returned in the structure [NET\\_DVR\\_ACS\\_WORK\\_STATUS\\_V50](#) by the output buffer (**IpOutBuffer**).



To check whether the device supports getting the working status of the access controller, you can call [NET\\_DVR\\_GetDeviceAbility](#), set the capability type **dwAbilityType** to "ACS\_ABILITY", and set the input parameter pointer **pInBuf** to the message [XML\\_Desc\\_AcsAbility](#) for getting the access control capability.

The capability is returned in the message [XML\\_AcsAbility](#) by the output parameter pointer **pOutBuf**. The related node is <AcsWorkStatus>.

## Log Mode

- Get configuration capability of log mode

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/LogModeCfg/capabilities?format=json**.

And the configuration capability is returned in the message **JSON\_Cap\_LogModeCfg** by the output parameter (**IpOutputParam**).

- Get log mode configuration parameters

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/LogModeCfg?format=json**.

And the parameters are returned in the message **JSON\_LogModeCfg** by the output buffer (**IpOutBuffer**) of the output parameter (**IpOutputParam**).

- Set log mode parameters

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/LogModeCfg?format=json** and set the input buffer (**IpInBuffer**) of the input parameter (**IpInputParam**) to the message **JSON\_LogModeCfg**.

## Event Optimization

- Get configuration capability of event optimization

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/EventOptimizationCfg/capabilities?format=json**.

And the configuration capability is returned in the message **JSON\_Cap\_EventOptimizationCfg** by the output parameter (**IpOutputParam**).

- Get event optimization configuration parameters

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/EventOptimizationCfg?format=json**.

And the parameters are returned in the message **JSON\_EventOptimizationCfg** by the output buffer (**IpOutBuffer**) of the output parameter (**IpOutputParam**).

- Set event optimization parameters

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/EventOptimizationCfg?format=json** and set the input buffer (**IpInBuffer**) of the input parameter (**IpInputParam**) to the message **JSON\_EventOptimizationCfg**.

## Active Infrared Intrusion Detection

- Get configuration capability of active infrared intrusion detection

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/Configuration/IRCfg/capabilities?format=json**.

And the configuration capability is returned in the message **JSON\_IRCfgCap** by **IpOutputParam**.

- Get parameters of active infrared intrusion detection

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/Configuration/IRCfg?format=json**.

And the parameters are returned in the message **JSON\_IRCfg** by **IpOutputParam**.

- Set parameters of active infrared intrusion detection

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/Configuration/IRCfg?format=json** and set the **IpInputParam** to **JSON\_IRCfg**.

## Card No. Authentication Mode

- Get configuration capability of card No. authentication mode

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/CardVerificationRule/capabilities?format=json**.

And the configuration capability is returned in the message **JSON\_CardVerificationRuleCap** by **IpOutputParam**.

- Get parameters of card No. authentication mode

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/CardVerificationRule?format=json**.

And the parameters are returned in the message **JSON\_CardVerificationRule** by **IpOutputParam**.

- Set parameters of card No. authentication mode

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/CardVerificationRule?format=json** and set the **IpInputParam** to the message **JSON\_CardVerificationRule**.

- Get switching progress and configuration result of card No. authentication mode

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/CardVerificationRule/progress?format=json**.

And the parameters are returned in the message **JSON\_CardVerificationRuleRes** by **IpOutputParam**.

## Additional Person Information

- Get configuration capability of name of additional person information

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/personInfoExtendName/capabilities?format=json**.

And the configuration capability is returned in the message **JSON\_PersonInfoExtendNameCap** by **IpOutputParam**.

- Get name of additional person information

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/personInfoExtendName?format=json**.

And the parameters are returned in the message **JSON\_PersonInfoExtendName** by **IpOutputParam**.

- Set name of additional person information

Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/personInfoExtendName?format=json** and set the **IpInputParam** to **JSON\_PersonInfoExtendName**.

## Privacy Settings

- Clear Pictures in Device

- Get the capability of clearing pictures in the device

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/ClearPictureCfg/capabilities?format=json](#).

And the capability is returned in the message [JSON\\_PersonInfoExtendNameCap](#) by [IpOutputParam](#).

- Clear pictures in the device

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: PUT [/ISAPI/AccessControl/ClearPictureCfg?format=json](#) and set the [IpInputParam](#) to [JSON\\_ClearPictureCfg](#).

- Configure Storage Parameters of Access Control Events

- Get the storage configuration capability of access control events

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/AcsEvent/StorageCfg/capabilities?format=json](#).

And the configuration capability is returned in the message [JSON\\_EventStorageCfgCap](#) by [IpOutputParam](#).

- Get the storage parameters of access control events

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/AcsEvent/StorageCfg?format=json](#).

And the parameters are returned in the message [JSON\\_EventStorageCfg](#) by [IpOutputParam](#).

- Set the storage parameters of access control events

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: PUT [/ISAPI/AccessControl/AcsEvent/StorageCfg?format=json](#) and set the [IpInputParam](#) to [JSON\\_EventStorageCfg](#).

## Health Code

- Health Code Settings

- Get the configuration capability of the health code

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/healthCodeCfg/capabilities?format=json](#).

And the capability is returned in the message [JSON\\_Cap\\_HealthCodeCfg](#) by [IpOutputParam](#).

- Get the health code parameters

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/healthCodeCfg?format=json](#).

And the parameters are returned in the message [JSON\\_HealthCodeCfg](#) by [IpOutputParam](#).

- Set the health code parameters

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: PUT [/ISAPI/AccessControl/healthCodeCfg?format=json](#) and set the [IpInputParam](#) to [JSON\\_HealthCodeCfg](#).

- Health Code Display

- Get the configuration capability of the health code display parameters

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/showHealthCodeCfg/capabilities?format=json](#).

And the capability is returned in the message [JSON\\_Cap\\_HealthCodeDisplayCfg](#) by [IpOutputParam](#).

- Get the health code display parameters

Call [NET\\_DVR\\_STDXMLConfig](#) to transmit the request URI: GET [/ISAPI/AccessControl/showHealthCodeCfg?format=json](#).

- And the parameters are returned in the message **JSON\_HealthCodeDisplayCfg** by **IpOutputParam**.
- Set the health code display parameters  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/showHealthCodeCfg?format=json** and set the **IpInputParam** to **JSON\_HealthCodeDisplayCfg**.

## Black Body

- Get the configuration capability of the black body  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/blackObject/capabilities?format=json**.  
And the capability is returned in the message **JSON\_Cap\_BlackBodyCfg** by **IpOutputParam**.
- Get the black body parameters  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/blackObject?format=json**.  
And the parameters are returned in the message **JSON\_BlackBodyCfg** by **IpOutputParam**.
- Set the black body parameters  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: PUT **/ISAPI/AccessControl/blackObject?format=json** and set the **IpInputParam** to **JSON\_BlackBodyCfg**.

## Getting Events Actively

- Get the capability of getting face temperature screening events actively  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/FaceTemperatureEvent/capabilities?format=json**.  
And the capability is returned in the message **JSON\_FaceTemperatureEventCap** by **IpOutputParam**.
- Get face temperature screening events actively  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: POST **/ISAPI/AccessControl/FaceTemperatureEvent?format=json** and set the **IpInputParam** to **JSON\_FaceTemperatureEventCond**.
- Get the capability of getting QR code scanning events actively  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/QRCodeEvent/capabilities?format=json**.  
And the capability is returned in the message **JSON\_QRCodeEventCap** by **IpOutputParam**.
- Get QR code scanning events actively  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: POST **/ISAPI/AccessControl/QRCodeEvent?format=json** and set the **IpInputParam** to **JSON\_QRCodeEventCond**.
- Get the capability of getting ID card swiping events actively  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: GET **/ISAPI/AccessControl/IDCardInfoEvent/capabilities?format=json**.  
And the capability is returned in the message **JSON\_IDCardInfoEventCap** by **IpOutputParam**.
- Get ID card swiping events actively  
Call **NET\_DVR\_STDXMLConfig** to transmit the request URI: POST **/ISAPI/AccessControl/IDCardInfoEvent?format=json** and set the **IpInputParam** to **JSON\_IDCardInfoEventCond**.

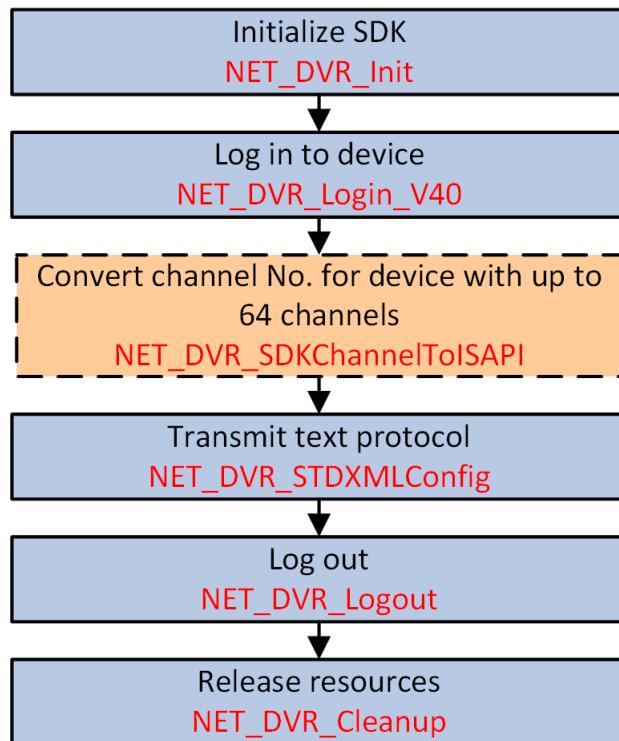
## 2.14 Integrate by Transmitting Text Protocol

The Device Network SDK support transmitting text protocol, including operation methods, request URIs, query parameters, and request or response messages, without any process between the platform or system and devices to extend the integration applications.

### Before You Start

- Make sure you have called [NET\\_DVR\\_Init](#) to initialize the programming environment.
- Make sure you have called [NET\\_DVR\\_Login\\_V40](#) to log in to the device.

### Steps



**Figure 1-24 API Calling Flow of Integrating by Transmitting Text Protocol**

1. **Optional:** Call [NET\\_DVR\\_SDKChannelToISAPI](#) to convert the device channel No. when integrating based on Device Network SDK and text protocol transmission.



- Note**
- This step is only available for rear-end devices with up to 64 network channels.
  - For the integration based on text protocol transmission, the channel No. starts from 1, for the integration based on Device Network SDK, the channel No. of device with up to 64 network channels starts from 33, so when the SDK's API is called by the platform or system for

transmitting text protocol to device, the channel No. returned by device starts from 1, but the start channel No. of platform or system should starts from 33, this may cause the problem.

2. Call **NET\_DVR\_STDXMLConfig** to transmit text protocol, including operation methods, request URLs, query parameters, and request or response messages, for realizing the corresponding applications.

### What to do next

Call **NET\_DVR\_Logout** and **NET\_DVR\_Cleanup** to log out and release resources.

## Chapter 3 API Reference

### 3.1 NET\_DVR\_Cleanup

Release the resources after the program is ended.

#### API Definition

```
BOOL NET_DVR_Cleanup()  
);
```

#### Return Values

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

If *FALSE* is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

The available error codes may be returned by this API are 0 and 3. See details in [\*\*Device Network SDK Errors\*\*](#).

#### Remarks

- When calling this API, you cannot call other APIs at the same time.
- [\*\*NET\\_DVR\\_Init\*\*](#) and this API should be called by pair. That is, once the [\*\*NET\\_DVR\\_Init\*\*](#) is called, you should call [\*\*NET\\_DVR\\_Cleanup\*\*](#) to release the resources when exiting the program.

### 3.2 NET\_DVR\_CloseAlarmChan\_V30

Close alarm uploading channel.

#### API Definition

```
BOOL NET_DVR_CloseAlarmChan_V30(  
    LONG    lAlarmHandle  
);
```

#### Parameters

##### IAlarmHandle

Value returned by [\*\*NET\\_DVR\\_SetupAlarmChan\\_V50\*\*](#).

#### Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

The available error codes of this API are 0, 3, 6, 12, 17, 41, and 47. See details in the [\*\*Device Network SDK Errors\*\*](#).

### 3.3 NET\_DVR\_ControlGateway

Call this API to remotely control the door or elevator.

#### API Definition

```
BOOL NET_DVR_ControlGateway(
    LONG     lUserID,
    LONG     lGatewayIndex,
    DWORD    dwStaic
);
```

#### Parameters

##### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

##### **lGatewayIndex**

[IN] Door No. or floor No., which starts from 1, -1: Control all doors or elevators of floors.

##### **dwStaic**

[IN] Command No.: 0-Close (Under Control), 1-Open, 2-Remain Open (Free), 3-Remain Closed (Disabled), 4-Recovery (only for elevator), 5-Vistor Call Elevator (only for elevator), 6-Resident Call Elevator (only for elevator).

#### Return Values

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

If returning failed, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

### 3.4 NET\_DVR\_GetDeviceAbility

Get the device capabilities.

#### API Definition

```
BOOL NET_DVR_GetDeviceAbility(
    LONG     lUserID,
    DWORD    dwAbilityType,
    char    *pInBuf,
    DWORD    dwInLength,
    char    *pOutBuf,
    DWORD    dwOutLength
);
```

## Parameters

### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

### **dwAbilityType**

[IN] Capability types, which are different according to different devices and functions.

### **pInBuf**

[IN] Input parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

### **dwInLength**

[IN] Size of input buffer.

### **pOutBuf**

[OUT] Output parameter buffer pointer, which are different according to different devices and functions, and they are returned in the structure or messages.

### **dwOutLength**

[OUT] Size of buffer for receiving data.

## Return Values

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

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.5 NET\_DVR\_GetDeviceConfig

Get device configuration information in batch (with sending data).

## API Definition

```
BOOL NET_DVR_GetDeviceConfig(
    LONG      lUserID,
    DWORD     dwCommand,
    DWORD     dwCount,
    LPVOID    lpInBuffer,
    DWORD     dwInBufferSize,
    LPVOID    lpStatusList,
    LPVOID    lpOutBuffer,
    DWORD     dwOutBufferSize
);
```

## Parameters

### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

## **dwCommand**

[IN] Device getting commands. The commands are different for different getting functions.

## **dwCount**

[IN] Number of configurations (cameras) to get at a time. 0, 1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 64 cameras' configuration information can be obtained at a time.

## **lpInBuffer**

[IN] Pointer of configuration condition buffer, which specifies the number (**dwCount**) of configurations to get, and relates to the getting commands.

## **dwInBufferSize**

[IN] Size of configuration condition buffer, which saves the obtained configuration information (the number is **dwCount**).

## **lpStatusList**

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras need to search, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0 or 1, it refers to getting succeeded, otherwise, this parameter value is the error code.

## **lpOutBuffer**

[OUT] Parameters returned by device, which relates to the getting commands. And there is a one-to-one correspondence between the parameters and the cameras need to search.

If the **lpStatusList** of one camera is larger than 1, the corresponding **lpOutBuffer** is invalid.

## **dwOutBufferSize**

[IN] Total size of returned results (the number is **dwCount**).

## **Return Values**

Returns *TRUE* for success, and returns *FALSE* for failure. If returns *TRUE*, it does not mean that all configurations are obtained, you can check the value of **lpStatusList[n]** to judge which one is succeeded.

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## **See Also**

[NET\\_DVR\\_SetDeviceConfig](#)

## **3.6 NET\_DVR\_GetDownloadState**

Get the file downloading progress and status.

## API Definition

```
LONG NET_DVR_GetDownloadState(
    LONG     lDownloadHandle,
    DWORD    *pProgress
);
```

## Parameters

### **lDownloadHandle**

[IN] Handle for downloading files, which is returned by [NET\\_DVR\\_StartDownload](#).

### **pProgress**

[OUT] Returned progress value, which is ranging from 1 to 100.

## Return Values

Returns -1 for calling failed, and returns other values as the downloading status codes: 1-Downloaded, 2-Downloading, 3-Downloading Failed, 4-Network Disconnected, Unknown Status. If returning failed, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.7 NET\_DVR\_GetDVRConfig

Get the device configuration information.

## API Definition

```
BOOL NET_DVR_GetDVRConfig(
    LONG     lUserID,
    DWORD    dwCommand,
    LONG     lRuleID,
    LONG     lChannel,
    LPVOID   lpOutBuffer,
    DWORD    dwOutBufferSize,
    LPDWORD  lpBytesReturned
);
```

## Parameters

### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

### **dwCommand**

[IN] Device getting commands, which are different according to different getting functions.

### **lRuleID**

[IN] Rule ID.

## IChannel

[IN] Channel No. (NIC No.), which varies with different commands. 0xffffffff-invalid or all channels, 1-main NIC, 2-extended NIC.

## IpOutBuffer

[OUT] Pointer of buffer to receive data. For different getting functions, the structures of this parameter are different.

## dwOutBufferSize

[IN] Size of buffer to receive data (unit: byte). It cannot be 0.

## lpBytesReturned

[OUT] Pointer of actually received data size. It cannot be NULL.

## Return Values

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

If *FALSE* is returned, you can call [\*\*\*NET\\_DVR\\_GetLastError\*\*\*](#) to get the error code.

The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the [\*\*\*Device Network SDK Errors\*\*\*](#).

## See Also

[\*\*\*NET\\_DVR\\_SetDVRConfig\*\*\*](#)

## 3.8 NET\_DVR\_GetErrorMsg

Return the error information of the last operation.

## API Definition

```
char *NET_DVR_GetErrorMsg(
    LONG    *pErrorNo
);
```

## Parameters

### pErrorNo

[OUT] Error code pointer.

## Return Values

The return values are the pointers of error information, see [\*\*\*Device Network SDK Errors\*\*\*](#) for details.

## Remarks

You can call [\*\*\*NET\\_DVR\\_GetLastError\*\*\*](#) to get the error codes.

## 3.9 NET\_DVR\_GetLastError

Return the error code of the last operation.

### API Definition

```
DWORD NET_DVR_GetLastError()  
);
```

### Return Values

The return values are error codes, see [Device Network SDK Errors](#) for details.

### Remarks

You can also call [NET\\_DVR\\_GetErrorMsg](#) to directly get the error information.

## 3.10 NET\_DVR\_GetNextRemoteConfig

Get the next search result.

### API Definition

```
LONG NET_DVR_GetNextRemoteConfig(  
    LONG     lHandle,  
    void     *lpOutBuff,  
    DWORD    dwOutBuffSize  
);
```

### Parameters

#### **lHandle**

[IN] Search handle, which is the value returned by [NET\\_DVR\\_StartRemoteConfig](#).

#### **lpOutBuff**

[OUT] Output parameter buffer pointer, which relates to the commands (**dwCommand**) of [NET\\_DVR\\_StartRemoteConfig](#).

#### **dwOutBuffSize**

[IN] Buffer size.

### Return Values

Returns -1 for failure, and returns other values for the current statuses, see details in the following table.

Status	Value	Description
NET_SDK_GET_NEXT_STATUS_SUCCESS	1000	The data is obtained. The API NET_DVR_GetNextRemoteConfig should be called again to get the next item of data.
NET_SDK_GET_NETX_STATUS_NEED_WAIT	1001	Waiting. The API NET_DVR_GetNextRemoteConfig can be called again.
NET_SDK_GET_NEXT_STATUS_FINISH	1002	All data is obtained. The API <b><u>NET_DVR_StopRemoteConfig</u></b> can be called to end.
NET_SDK_GET_NEXT_STATUS_FAILED	1003	Getting data exception. The API <b><u>NET_DVR_StopRemoteConfig</u></b> can be called to end.

If -1 is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

## Remarks

To get all information, you should call this API repeatedly.

## 3.11 NET\_DVR\_GetSDKLocalCfg

Get the HCNetSDK's local configuration parameters.

### API Definition

```
BOOL NET_DVR_GetSDKLocalCfg(
    NET_SDK_LOCAL_CFG_TYPE     enumType,
    void                      *lpOutBuff
);
```

### Parameters

#### enumType

[IN] Configuration options. Different values of configuration options correspond to different parameters, see details in **NET\_SDK\_LOCAL\_CFG\_TYPE**.

#### lpOutBuff

[OUT] Output parameters. For different configuration options, the structures of output parameters are different, see details in **NET\_SDK\_LOCAL\_CFG\_TYPE**.

### Return Values

Returns *TRUE* for success, and returns *FALSE* for failure. If *FALSE* is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

## See Also

[NET\\_DVR\\_SetSDKLocalCfg](#)

## 3.12 NET\_DVR\_GetUploadState

Get the file uploading progress and status.

### API Definition

```
LONG NET_DVR_GetUploadState(
    LONG      lUploadHandle,
    DWORD     *pProgress
) ;
```

### Parameters

#### **lUploadHandle**

[IN] Handling for uploading files, which is returned by [NET\\_DVR\\_UploadFile\\_V40](#).

#### **pProgress**

[OUT] Returned progress value.

### Return Values

Return -1 for failure, and return other values as the uploading status codes, see details in the following table.

**Table 2-1 Uploading Status Code**

Return Value	Description
1	Uploaded successfully.
2	Uploading.
3	Uploading failed.
4	Network disconnected. Unknown status.
6	HDD error.
7	No HDD for saving inquest files.
8	Insufficient capacity.
9	Insufficient device resource.
10	No more files can be uploaded.
11	Too large file size.

Return Value	Description
15	File type error.
19	Invalid file format.
20	Incorrect file content.
21	The uploaded audio sampling rate is not supported.
22	Insufficient storage in the face library.
26	Name error.
27	Invalid picture resolution.
28	Too many targets on the picture.
29	No target is recognized on the picture.
30	Picture recognition failed.
31	Analysis engine exception.
32	Analyzing additional information on the picture failed.
33	Thumbnail modeling failed.
34	Incorrect security verification key.
35	Downloading picture via URL has not started.
36	Duplicate custom ID of different persons.
37	Person ID error (The ID is saved in <b>customHumanID</b> of <b>FaceAppendData</b> ).
38	Modeling failed. Device inner error.
39	Modeling failed. Face modeling error.
40	Modeling failed. Face score error.
41	Modeling failed. Feature collection error.
42	Modeling failed. Attribute collection error.
43	Picture data error.
44	Picture additional information error.
45	Certificate has already existed.

### 3.13 NET\_DVR\_Init

Initialize the programming environment before calling other APIs.

#### API Definition

```
BOOL NET_DVR_Init()  
);
```

#### Return Values

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

If *FALSE* is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

The available error codes of this API are 0, 41, and 53. See details in [\*\*Device Network SDK Errors\*\*](#).

#### Remarks

Before initializing, you can call [\*\*NET\\_DVR\\_SetSDKInitCfg\*\*](#) to set the initialization parameters, such as supported capabilities, loading path of component libraries (only supported by Linux system), and so on.

#### See Also

[\*\*NET\\_DVR\\_Cleanup\*\*](#)

### 3.14 NET\_DVR\_Login\_V40

Log in to the device (supports asynchronous login).

#### API Definition

```
LONG NET_DVR_Login_V40(  
    NET_DVR_USER_LOGIN_INFO    pLoginInfo,  
    NET_DVR_DEVICEINFO_V40     lpDeviceInfo  
) ;
```

#### Parameters

##### pLoginInfo

[IN] Login parameters, including device address, user name, password, and so on. See details in the structure [\*\*NET\\_DVR\\_USER\\_LOGIN\\_INFO\*\*](#).

##### lpDeviceInfo

[OUT] Device information. See details in the structure [\*\*NET\\_DVR\\_DEVICEINFO\\_V40\*\*](#).

## Return Values

- For asynchronous login, the callback function (**fLoginResultCallBack**) configured in the structure (**NET\_DVR\_USER\_LOGIN\_INFO**) returns the asynchronous login status, user ID and device information.
- For synchronous login, this API returns -1 for logging failed, and returns other values for the returned user IDs. The user ID is unique, and it helps to realize the further device operations.
- If -1 is returned, you can call **NET\_DVR\_GetLastError** to get the error code.

## Remarks

- When **bUseAsynLogin** in **pLoginInfo** is 0, it indicates that login is in synchronous mode; when **bUseAsynLogin** in **pLoginInfo** is 1, it indicates that login is in asynchronous mode.
- Up to 2048 users are allowed to log in to HCNetSDK at same time, and the values of returned **UserID** are ranging from 0 to 2047.

## See Also

[\*\*NET\\_DVR\\_Logout\*\*](#)

## 3.15 NET\_DVR\_Logout

Log out from devices.

### API Definitions

```
BOOL NET_DVR_Logout (
    LONG     lUserID
);
```

### Parameters

#### **lUserID**

[IN] User ID, which is returned by [\*\*NET\\_DVR\\_Login\\_V40\*\*](#).

## Return Values

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

If *FALSE* is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

The available error codes may be returned by this API are 0, 3, 7, 8, 9, 10, 14, 17, 41, 44, 47, 72, and 73. See details in [\*\*Device Network SDK Errors\*\*](#).

## 3.16 NET\_DVR\_SDKChannelToISAPI

Convert channel No. between the private protocol and a text protocol.

### API Definition

```
BOOL NET_DVR_SDKChannelToISAPI (
    LONG     lUserID,
    LONG     lInChannel,
    BOOL     bSDKToISAPI
);
```

### Parameters

#### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

#### **lInChannel**

[IN] Channel No.

#### **bSDKToISAPI**

[OUT] Channel No. conversion type: "TRUE"-convert channel No. of private protocol to that of text protocol, "FALSE"-convert channel No. of text protocol to that of private protocol.

### Return Values

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

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.17 NET\_DVR\_SendRemoteConfig

Send data via the persistent connection.

### API Definition

```
BOOL NET_DVR_SendRemoteConfig (
    LONG     lHandle,
    DWORD    dwDataType,
    char    *pSendBuf,
    DWORD    dwBufSize
);
```

### Parameters

#### **lHandle**

Persistent configuration handle, which is returned by [NET\\_DVR\\_StartRemoteConfig](#).

#### **dwDataType**

[IN] Data type, which relates to the commands of [NET\\_DVR\\_StartRemoteConfig](#).

#### **pSendBuf**

[IN] Buffer for saving data to be sent, which relates to **dwDataType**.

## **dwBufSize**

[IN] Size of data to be sent.

## **Return Values**

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

If *FALSE* is returned, you can call [\*\*\*NET\\_DVR\\_GetLastError\*\*\*](#) to get the error code.

## **Remarks**

Before calling this API, you must call [\*\*\*NET\\_DVR\\_StartRemoteConfig\*\*\*](#) to get the persistent connection handle.

## **3.18 NET\_DVR\_SetConnectTime**

Set network connection timeout and connection attempts.

## **API Definition**

```
BOOL NET_DVR_SetConnectTime (
    DWORD   dwWaitTime,
    DWORD   dwTryTimes
);
```

## **Parameters**

### **dwWaitTime**

[IN] Timeout, unit: ms, value range: [300,75000]; the maximum timeout varies with different operating systems.

### **dwTryTimes**

[IN] Connection attempts (reserved).

## **Return Values**

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call [\*\*\*NET\\_DVR\\_GetLastError\*\*\*](#) to get the error code.

## **Remarks**

- For Windows operating system, the default connection timeout is 3000 ms; for Linux operating system with version 5.2.7.2 and above, the default connection timeout is 3500 ms.
- For HCNetSDK with version 4.0 and above, when the configured timeout is larger than or smaller than the limit value, this API will not return *FALSE*, it will automatically use the timeout that is closest to the limit value as the actual timeout.

### 3.19 NET\_DVR\_SetDeviceConfig

Set device parameters in batch (sending data is supported).

#### API Definition

```
BOOL NET_DVR_SetDeviceConfig(
    LONG      lUserID,
    DWORD     dwCommand,
    DWORD     dwCount,
    LPVOID    lpInBuffer,
    DWORD     dwInBufferSize,
    LPVOID    lpStatusList,
    LPVOID    lpInParamBuffer,
    DWORD     dwInParamBufferSize
);
```

#### Parameters

##### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

##### **dwCommand**

[IN] Device configuration commands, which are different according to different configurations.

##### **dwCount**

[IN] Number of cameras to be set at a time. 0,1-one camera, 2-two cameras, 3-three cameras, and so on. Up to 256 cameras can be configured at a time.

##### **lpInBuffer**

[IN] Pointer of configuration condition buffer, e.g., stream ID, which specifies the number (**dwCount**) of cameras to set, and relates to the configuration commands.

##### **dwInBufferSize**

[IN] Size of configuration condition buffer, which saves the configured information of cameras with the number of **dwCount**.

##### **lpStatusList**

[OUT] Error information list, and its memory is allocated by user, each error information contains 4 bytes (a unsigned 32-bit integer).

There is a one-to-one correspondence between the errors in the list and the cameras that need to be searched, e.g., **lpStatusList[2]** corresponds to **lpInBuffer[2]**.

If the parameter value is 0, it refers to setting succeeded, otherwise, this parameter value is the error code.

##### **lpInParamBuffer**

[IN] Device parameters to set, which relates to the configuration commands. And there is a one-to-one correspondence between the parameters and the cameras that need to be searched.

#### **dwInParamBufferSize**

[IN] Set the size of content buffer.

#### **Return Values**

Returns *TRUE* for success, and returns *FALSE* for all failed. If returns *TRUE*, it does not indicate that all settings are succeeded, you can get the value of **IpStatusList[n]** to check which one is succeeded.

If *FALSE* is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

#### **See Also**

[\*\*NET\\_DVR\\_GetDeviceConfig\*\*](#)

## **3.20 NET\_DVR\_SetDVRConfig**

Set the device parameters.

#### **API Definition**

```
BOOL NET_DVR_SetDVRConfig(
    LONG     lUserID,
    DWORD    dwCommand,
    LONG     lChannel,
    LPVOID   lpInBuffer,
    DWORD    dwInBufferSize
);
```

#### **Parameters**

##### **lUserID**

[IN] Value returned by [\*\*NET\\_DVR\\_Login\\_V40\*\*](#).

##### **dwCommand**

[IN] Device configuration commands, which are different according to different configuration functions.

##### **lChannel**

[IN] Channel No. (NIC No.), which varies with different commands. 0xFFFFFFFF-invalid, 1-main NIC, 2-extended NIC.

##### **lpInBuffer**

[IN] Pointer of input data buffer. For different configuration functions, the structures of this parameter are different.

##### **dwInBufferSize**

[IN] Size of input data buffer (unit: byte).

### Return Values

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

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

The following error codes may be returned by this API: 0, 3, 6, 7, 8, 9, 10, 12, 17, 41, 43, 44, 47, 72, 73, and 76. See the corresponding error types and descriptions in the [Device Network SDK Errors](#).

### See Also

[NET\\_DVR\\_GetDVRConfig](#)

## 3.21 NET\_DVR\_SetDVRMessageCallBack\_V50

Set callback functions for getting the video data.

### API Definition

```
BOOL NET_DVR_SetDVRMessageCallBack_V50 (
    int          iIndex,
    MSGCallBack  fMessageCallBack,
    void         *pUser
);
```

### Parameters

#### iIndex

[IN] Callback function index No., which ranges from 0 to 15.

#### fMessageCallBack

[IN] Callback function, see details in [MSGCallBack](#).

#### pUser

[IN] User data.

### Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* returned, call [NET\\_DVR\\_GetLastError](#) to get the error code.

### Remarks

- This API supports setting multiple callback functions for different channels (up to 16 channels are supported) at same time, and the configured callback functions are distinguished by the index No.
- All alarm/event information will be returned in each configured callback function, and you can distinguish the devices via the **pAlarmInfo** in the callback function ([MSGCallBack](#)).

### Example

Sample Code of Setting Multiple Callback Functions to Receive Different Alarms/Events in Arming Mode

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

int iNum=0;
void CALLBACK MessageCallbackNo1(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void* pUser)
{
    int i=0;
    char filename[100];
    FILE *fSnapPic=NULL;
    FILE *fSnapPicPlate=NULL;

    //This sample code is for reference only. Actually, it is not recommended
    to process the data and save file in the callback function directly.
    //You'd better process the data in the message response funcion via message
    mode (PostMessage) .

    switch(lCommand)
    {
        case COMM_ALARM:
        {
            NET_DVR_ALARMINFO struAlarmInfo;
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
            switch (struAlarmInfo.dwAlarmType)
            {
                case 3: //Motion detection alarm
                    for (i=0; i<16; i++)    //#define MAX_CHANNUM 16 //The
maximum number of channels
                    {
                        if (struAlarmInfo.dwChannel[i] == 1)
                        {
                            printf("Channel Number with Motion Detection Alarm
%d\n", i+1);
                        }
                    }
                    break;
                default:
                    break;
            }
            break;
        }
        case COMM_UPLOAD_PLATE_RESULT:
        {
            NET_DVR_PLATE_RESULT struPlateResult={0};
            memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
        }
    }
}
```

```

        printf("License Plate Number: %s\n",
struPlateResult.struPlateInfo.sLicense);//License plate number

        switch(struPlateResult.struPlateInfo.byColor)//License plate color
        {
        case VCA_BLUE_PLATE:
            printf("Vehicle Color: Blue\n");
            break;
        case VCA_YELLOW_PLATE:
            printf("Vehicle Color: Yellow\n");
            break;
        case VCA_WHITE_PLATE:
            printf("Vehicle Color: White\n");
            break;
        case VCA_BLACK_PLATE:
            printf("Vehicle Color: Black\n");
            break;
        default:
            break;
        }
        //Scene picture
        if (struPlateResult.dwPicLen != 0 && struPlateResult.byResultType
== 1 )
        {
            sprintf(filename,"testpic_%d.jpg",iNum);
            fSnapPic=fopen(filename,"wb");
            fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,
1,fSnapPic);
            iNum++;
            fclose(fSnapPic);
        }
        //License plate picture
        if (struPlateResult.dwPicPlateLen != 0 &&
struPlateResult.byResultType == 1)
        {
            sprintf(filename,"testPicPlate_%d.jpg",iNum);
            fSnapPicPlate=fopen(filename,"wb");
            fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,
1,fSnapPicPlate);
            iNum++;
            fclose(fSnapPicPlate);
        }
        //Processing other data...
        break;
    }
    case COMM_ITS_PLATE_RESULT:
    {
        NET_ITS_PLATE_RESULT struITSPlateResult={0};
        memcpy(&struITSPlateResult, pAlarmInfo, sizeof(struITSPlateResult));

        for (i=0;i<struITSPlateResult.dwPicNum;i++)
        {

```

```

        printf("License Plate Number: %s\n",
struITSPlateResult.struPlateInfo.sLicense);//License plate number
        switch(struITSPlateResult.struPlateInfo.byColor)//License plate
color
        {
            case VCA_BLUE_PLATE:
                printf("Vehicle Color: Blue\n");
                break;
            case VCA_YELLOW_PLATE:
                printf("Vehicle Color: Yellow\n");
                break;
            case VCA_WHITE_PLATE:
                printf("Vehicle Color: White\n");
                break;
            case VCA_BLACK_PLATE:
                printf("Vehicle Color: Black\n");
                break;
            default:
                break;
        }
        //Save scene picture
        if ((struITSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struITSPlateResult.struPicInfo[i].byType== 1)|| (struITSPlateResult.struPicInfo[i].byType == 2))
        {
            sprintf(filename,"testITSpic%d_%d.jpg",iNum,i);
            fSnapPic=fopen(filename,"wb");
            fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen,1,fSnapPic);
            iNum++;
            fclose(fSnapPic);
        }
        //License plate thumbnails
        if ((struITSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struITSPlateResult.struPicInfo[i].byType == 0))
        {
            sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
            fSnapPicPlate=fopen(filename,"wb");
            fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen, 1, \ fSnapPicPlate);
            iNum++;
            fclose(fSnapPicPlate);
        }
        //Processing other data...
    }
    break;
}
default:
    break;
}
}

```

```

void CALLBACK MessageCallbackNo2(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char
*pAlarmInfo, DWORD dwBufLen, void* pUser)
{
    int i=0;
    char filename[100];
    FILE *fSnapPic=NULL;
    FILE *fSnapPicPlate=NULL;

    //This sample code is for reference only. Actually, it is not recommended
    to process the data and save file in the callback function directly.
    //You'd better process the data in the message response funcion via message
    mode (PostMessage) .

    switch(lCommand)
    {
        case COMM_ALARM:
        {
            NET_DVR_ALARMINFO struAlarmInfo;
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
            switch (struAlarmInfo.dwAlarmType)
            {
                case 3: //Motion detection alarm
                    for (i=0; i<16; i++) //#define MAX_CHANNUM 16 //The
maximum number of channel
                    {
                        if (struAlarmInfo.dwChannel[i] == 1)
                        {
                            printf("Channel No. with Motion Detection Alarm %d
\n", i+1);
                        }
                    }
                    break;
                default:
                    break;
            }
            break;
        }
        case COMM_UPLOAD_PLATE_RESULT:
        {
            NET_DVR_PLATE_RESULT struPlateResult={0};
            memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
            printf("License Plate Number: %s\n",
struPlateResult.struPlateInfo.sLicense);//License plate number

            switch(struPlateResult.struPlateInfo.byColor)//License plate color
            {
                case VCA_BLUE_PLATE:
                    printf("Vehicle Color: Blue\n");
                    break;
                case VCA_YELLOW_PLATE:
                    printf("Vehicle Color: Yellow\n");
                    break;
            }
        }
    }
}

```

```

        case VCA_WHITE_PLATE:
            printf("Vehicle color: White\n");
            break;
        case VCA_BLACK_PLATE:
            printf("Vehicle Color: Black\n");
            break;
        default:
            break;
    }
    //Scene picture
    if (struPlateResult.dwPicLen != 0 && struPlateResult.byResultType
== 1 )
    {
        sprintf(filename,"testpic_%d.jpg",iNum);
        fSnapPic=fopen(filename,"wb");
        fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,
1,fSnapPic);
        iNum++;
        fclose(fSnapPic);
    }
    //License plate picture
    if (struPlateResult.dwPicPlateLen != 0 &&
struPlateResult.byResultType == 1)
    {
        sprintf(filename,"testPicPlate_%d.jpg",iNum);
        fSnapPicPlate=fopen(filename,"wb");
        fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,
1,fSnapPicPlate);
        iNum++;
        fclose(fSnapPicPlate);
    }
    //Processing other data...
    break;
}
case COMM_ITS_PLATE_RESULT:
{
    NET_ITS_PLATE_RESULT struITSPlateResult={0};
    memcpy(&struITSPlateResult, pAlarmInfo, sizeof(struITSPlateResult));

    for (i=0;i<struITSPlateResult.dwPicNum;i++)
    {
        printf("License Plate Number: %s\n",
struITSPlateResult.struPlateInfo.sLicense);//License plate number
        switch(struITSPlateResult.struPlateInfo.byColor)//License plate
color
        {
            case VCA_BLUE_PLATE:
                printf("Vehicle Color: Blue\n");
                break;
            case VCA_YELLOW_PLATE:
                printf("Vehicle Color: Yellow\n");
                break;
        }
    }
}

```

```

        case VCA_WHITE_PLATE:
            printf("Vehicle Color: White\n");
            break;
        case VCA_BLACK_PLATE:
            printf("Vehicle Color: Black\n");
            break;
        default:
            break;
    }
    //Save scene picture
    if ((struITSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struITSPlateResult.struPicInfo[i].byType== 1)|| (struITSPlateResult.struPicInfo[i].byType == 2))
    {
        sprintf(filename,"testITSpic%d_%d.jpg",iNum,i);
        fSnapPic=fopen(filename,"wb");
        fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen,1,fSnapPic);
        iNum++;
        fclose(fSnapPic);
    }
    //License plate thumbnails
    if ((struITSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struITSPlateResult.struPicInfo[i].byType == 0))
    {
        sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
        fSnapPicPlate=fopen(filename,"wb");
        fwrite(struITSPlateResult.struPicInfo[i].pBuffer,
struITSPlateResult.struPicInfo[i].dwDataLen, 1, \ fSnapPicPlate);
        iNum++;
        fclose(fSnapPicPlate);
    }
    //Processing other data...
}
break;
}
default:
    break;
}
}

void main() {

//-----
//Initialize
NET_DVR_Init();
//Set the connection time and reconnection time
NET_DVR_SetConnectTime(2000, 1);
NET_DVR_SetReconnect(10000, true);

//-----
//Log in to device

```

```
LONG lUserID;
NET_DVR_DEVICEINFO_V30 struDeviceInfo;
lUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345",
&struDeviceInfo);
if (lUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//Set alarm callback function
NET_DVR_SetDVRMessageCallBack_V50(0, MessageCallbackNo1, NULL);
NET_DVR_SetDVRMessageCallBack_V50(1, MessageCallbackNo2, NULL);

//Enable arming
NET_DVR_SETUPALARMPARAM struSetupParam={0};
struSetupParam.dwSize=sizeof(NET_DVR_SETUPALARMPARAM);

//Alarm information type to upload: 0-History Alarm (NET_DVR_PLATE_RESULT), 1-
Real-Time Alarm (NET_ITS_PLATE_RESULT)
struSetupParam.byAlarmInfoType=1;
//Arming Level: Level-2 arming (for traffic device)
struSetupParam.byLevel=1;

LONG lHandle = NET_DVR_SetupAlarmChan_V41(lUserID,&struSetupParam);
if (lHandle < 0)
{
    printf("NET_DVR_SetupAlarmChan_V41 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(20000);
//Disarm uploading channel
if (!NET_DVR_CloseAlarmChan_V30(lHandle))
{
    printf("NET_DVR_CloseAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//User logout
NET_DVR_Logout(lUserID);
//Release SDK resource
NET_DVR_Cleanup();
return;
}
```

## See Also

[NET\\_DVR\\_SetupAlarmChan\\_V50](#)

## 3.22 NET\_DVR\_SetSDKInitCfg

Set initialization parameters.

### API Parameters

```
BOOL NET_DVR_SetSDKInitCfg (
    NET_SDK_INIT_CFG_TYPE      enumType,
    void* const                lpInBuff
);
```

### Parameters

#### enumType

[IN] Initialization parameter type. Different type values correspond to different parameters, see details in the table below.

Table 2-2 NET\_SDK\_INIT\_CFG\_TYPE

enumType	Value	Description	lpInBuff
NET_SDK_INIT_CFG_ABILITY	1	Capability supported by SDK.	<a href="#"><u>NET_DVR_INIT_CFG_ABILITY</u></a>
NET_SDK_INIT_CFG_SDK_PATH	2	Set loading path for component libraries (supported by both Linux and Windows system).	<a href="#"><u>NET_DVR_LOCAL_SDK_PATH</u></a>
NET_SDK_INIT_CFG_LIBEAY_PATH	3	Set path (including library name) for libeay32.dll (Windows), libcrypto.so (Linux), and libcrypto.dylib (Mac) of OpenSSL in version 1.1.1 and 1.0.2.	Path in string format, e.g., <b>C:\libeay32.dll</b> .
NET_SDK_INIT_CFG_SSLEAY_PATH	4	Set path (including library name) for ssleay32.dll (Windows), libssl.so (Linux),	Path in string format, e.g., <b>C:\ssleay32.dll</b> .

enumType	Value	Description	lpInBuff
		libssl.dylib (Mac) or OpenSSL in version 1.1.1 and 1.0.2.	

**lpInBuff**

[IN] Input parameter. Different parameter types correspond to different structures, see details in the table above.

**Return Values**

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

If *FALSE* is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

**Remarks**

This API should be called before calling [\*\*NET\\_DVR\\_Init\*\*](#) to initialize and check the dependent libraries or capabilities. This API only takes effect for POSIX. For Windows, it takes no effect but success will be returned.

### 3.23 NET\_DVR\_SetSDKLocalCfg

Set the local parameters.

**API Definition**

```
BOOL NET_DVR_SetSDKLocalCfg(
    NET_SDK_LOCAL_CFG_TYPE      enumType,
    void* const                 lpInBuff
);
```

**Parameters****enumType**

[IN] Configuration options. Different values of configuration options correspond to different SDK parameters, see details in [\*\*NET\\_SDK\\_LOCAL\\_CFG\\_TYPE\*\*](#).

**lpInBuff**

[IN] Input parameters. For different configuration options, the structures of input parameters are different, see details in [\*\*NET\\_SDK\\_LOCAL\\_CFG\\_TYPE\*\*](#).

**Return Values**

Returns *TRUE* for success, and returns *FALSE* for failure. If *FALSE* is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

Before setting parameters for this function, make sure no device has logged in.

## See Also

[NET\\_DVR\\_GetSDKLocalCfg](#)

## 3.24 NET\_DVR\_SetupAlarmChan\_V50

Set up persistent connection to receive alarm/event information (supports alarm/event subscription).

### API Definition

```
LONG NET_DVR_SetupAlarmChan_V50(
    LONG          IUserID,
    NET_DVR_SETUPALARM_PARAM_V50      lpSetupParam,
    char           *pData,
    DWORD          dwDataLen,
) ;
```

### Parameters

#### **IUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

#### **lpSetupParam**

[IN] Arming parameters, refer to the structure [NET\\_DVR\\_SETUPALARM\\_PARAM\\_V50](#) for details.

#### **pData**

[IN] Alarm/event subscription conditions.

#### **dwDataLen**

[IN] Length of alarm/event subscription conditions.

### Return Values

Return -1 for failure, and return other values as the handles of [NET\\_DVR\\_CloseAlarmChan\\_V30](#). If -1 is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

### Remarks

This API supports alarm/event subscription, you can specify the types of alarm or event to be uploaded by device by setting **pData** and **dwDataLen**.

## 3.25 NET\_DVR\_StartDownload

Start downloading files

## API Definition

```
LONG NET_DVR_StartDownload(
    LONG          lUserID,
    DWORD         dwDownloadType,
    LPVOID        lpInBuffer,
    DWORD         dwInBufferSize,
    char const   *sFileName
);
```

## Parameters

### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

### **dwDownloadType**

[IN] Downloading commands which specify the file type to download, see details in the enumeration [NET\\_SDK\\_DOWNLOAD\\_TYPE](#).

### **lpInBuffer**

[IN] Input parameters, which are different according to different downloading commands.

### **dwInBufferSize**

[IN] Input buffer size.

### **sFileName**

[IN] Path for saving downloaded files (absolute path, includes file name).

## Return Values

Returns -1 for failure, and returns other values as the parameters of [NET\\_DVR\\_StopDownload](#) and [NET\\_DVR\\_GetDownloadState](#).

If returning failed, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.26 NET\_DVR\_StartListen\_V30

Register callback function for receiving alarm/event information and start listening (supports multiple threads).

## API Definition

```
LONG NET_DVR_StartListen_V30(
    char          *sLocalIP,
    WORD          wLocalPort,
    MSGCallBack  DataCallback,
    void          *pUserData
);
```

### Parameters

#### sLocalIP

[IN] IP address of local PC. It can be set to null.

#### wLocalPort

[IN] Listening port No. of local PC. It is configured by user, and it should be the same with that of device.

#### DataCallback

[IN] Alarm/event information callback function, see details in [\*\*MSGCallBack\*\*](#).

#### pUserData

[IN] User data.

### Return Values

Return -1 for failure, and return other values for the handle parameters of

#### [\*\*NET\\_DVR\\_StopListen\\_V30\*\*](#)

If -1 is returned, you can call [\*\*NET\\_DVR\\_GetLastError\*\*](#) to get the error code.

The available error codes of this API are 0, 3, 6, 12, 17, 41, 44, 47, 72, and 75. See details in the [\*\*Device Network SDK Errors\*\*](#).

### Remarks

- To receive the alarm/event information sent by device, you should set the management host server address or listening host server address of device to the IP address of PC (which is same with the **sLocalIP**), or set the management host server port or listening host server port to the listening port No. of PC (which is same with the **wLocalPort**).
- The callback function in this API is prior to other callback functions, that is, if the callback function is configured in this API, other callback functions will not receive the alarm information. All the device alarm information is returned in same callback function, and you can distinguish the devices via the alarm device information (**pAlarmInfo**).

## 3.27 NET\_DVR\_StartRemoteConfig

Enable remote configuration.

### API Definition

```
LONG NET_DVR_StartRemoteConfig(
    LONG          lUserID,
    DWORD         dwCommand,
    LPVOID        lpInBuffer,
    DWORD         dwInBufferLen,
    fRemoteConfigCallback cbStateCallback,
```

```
LPVOID pUserData  
) ;
```

## Parameters

### **IUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

### **dwCommand**

[IN] Configuration commands. For different functions, the commands and **IpInBuffer** are different, see the detailed relation in the table below:

<b>dwCommand Macro Definition</b>	<b>Value</b>	<b>Description</b>	<b>IpInBuffer Related Structure</b>	<b>IpBuffer Related Structure</b>
NET_DVR_GET_ALL_RECORD_PASSBACK_TASK_MANUAL	6235	Get tasks of manually copying back videos	<a href="#"><u>NET_DVR_RECO_RD_PASSBACK_MANUAL_COND</u></a>	<a href="#"><u>NET_DVR_RECORDDPASSBACKMANUALCONDRET</u></a>

### **IpInBuffer**

Input parameter buffer pointer, which relates to the configuration command.

### **dwInBufferLen**

[IN] Size of input buffer.

### **cbStateCallback**

[IN] Status callback function, see the definition in [fRemoteConfigCallback](#).

### **pUserData**

[OUT] User data.

## Return Values

Returns -1 for failure, and returns other values for the handles of [NET\\_DVR\\_GetNextRemoteConfig](#) and [NET\\_DVR\\_StopRemoteConfig](#).

If -1 is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## Remarks

This API specifies the information to search. After calling this API, you can call [NET\\_DVR\\_GetNextRemoteConfig](#) to get the information one by one.

## 3.28 NET\_DVR\_STDXMLConfig

Transmit request URL with XML or JSON format to implement some typical functions.

## API Definition

```
BOOL NET_DVR_STDXMLConfig(
    LONG                                     lUserID,
    const NET_DVR_XML_CONFIG_INPUT          *lpInputParam,
    NET_DVR_XML_CONFIG_OUTPUT               *lpOutputParam
);
```

## Parameters

### **lUserID**

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

### **lpInputParam**

[IN] Input parameters, refer to the structure [NET\\_DVR\\_XML\\_CONFIG\\_INPUT](#) for details.

### **lpOutputParam**

[IN][OUT] Output parameters, refer to the structure [NET\\_DVR\\_XML\\_CONFIG\\_OUTPUT](#) for details.

## Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## Remarks

The input parameter **lpInputParam** and output parameter **lpOutputParam** are different when transmitting text protocol for implementing different functions, and each parameter corresponds to a component of text protocol, see the relations below:

<b>Parameter of NET_DVR_STDXMLConfig</b>		<b>Component of Text Protocol</b>
<b>lpInputParam</b>	<b>lpRequestMethod</b> (see in structure <a href="#"><u>NET_DVR_XML_CONFIG_INPUT</u></a> )	Method+URL E.g., GET /ISAPI/System/ capabilities
	<b>lpInBuffer</b> (see in structure <a href="#"><u>NET_DVR_XML_CONFIG_INPUT</u></a> )	Request Message
<b>lpOutputParam</b>	<b>lpOutBuffer</b> (see in structure <a href="#"><u>NET_DVR_XML_CONFIG_OUTPUT</u></a> )	Response Message
	<b>lpStatusBuffer</b> (see in structure <a href="#"><u>NET_DVR_XML_CONFIG_OUTPUT</u></a> )	Response Message

## 3.29 NET\_DVR\_StopDownload

Stop downloading files.

### API Definition

```
BOOL NET_DVR_StopDownload(
    LONG   lHandle
);
```

### Parameters

#### lHandle

[IN] Handle for downloading files, which is returned by [NET\\_DVR\\_StartDownload](#).

### Return Values

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

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.30 NET\_DVR\_StopListen\_V30

Stop listening (supports multiple threads).

### API Definition

```
BOOL NET_DVR_StopListen_V30(
    LONG   lListenHandle
);
```

### Parameters

#### lListenHandle

Listening handle, which is returned by [NET\\_DVR\\_StartListen\\_V30](#).

### Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

The available error codes of this API are 0, 3, 12, and 17. See details in the [Device Network SDK Errors](#).

## 3.31 NET\_DVR\_StopRemoteConfig

Disconnect the persistent connection to stop remote configuration, and release resources.

### API Definition

```
BOOL NET_DVR_StopRemoteConfig(
    LONG     lHandle
);
```

### Parameters

#### lHandle

[IN] Handle, which is returned by [NET\\_DVR\\_StartRemoteConfig](#).

### Return Values

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

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.32 NET\_DVR\_UploadClose

Stop uploading files.

### API Definition

```
BOOL NET_DVR_UploadClose(
    LONG     lUploadHandle
);
```

### Parameters

#### lUploadHandle

[IN] Handle for uploading files, which is returned by [NET\\_DVR\\_UploadFile\\_V40](#).

### Return Values

Return *TRUE* for success, and return *FALSE* for failure.

If *FALSE* is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.33 NET\_DVR\_UploadFile\_V40

Upload file.

### API Definition

```
LONG NET_DVR_UploadFile_V40 (
    LONG     lUserID,
    DWORD    dwUploadType,
    LPVOID   lpInBuffer,
    DWORD    dwInBufferSize,
```

```
char      *sFileName,
LPVOID   lpOutBuffer,
DWORD    dwOutBufferSize
);
```

### Parameters

#### IUserID

[IN] Value returned by [NET\\_DVR\\_Login\\_V40](#).

#### dwUploadType

[IN] Uploading commands, which specify the file type to upload, see details in the enumeration [NET\\_SDK\\_UPLOAD\\_TYPE](#).

#### lpInBuffer

[IN] Input parameters, which are different according to different uploading commands.

#### dwInBufferSize

[IN] Input buffer size.

#### sFileName

[IN] Name of the file to be uploaded. For the complete file path (including the file name), the maximum size is 128 bytes, and the maximum size of the file name is 32 bytes.

#### lpOutBuffer

[OUT] Output parameters, which are different according to different uploading commands.

#### dwOutBufferSize

[OUT] Output buffer size.

### Return Values

Return -1 for failure, and return other values as the parameter of [NET\\_DVR\\_UploadClose](#) and [NET\\_DVR\\_GetUploadState](#).

If -1 is returned, you can call [NET\\_DVR\\_GetLastError](#) to get the error code.

## 3.34 Callback Function

### 3.34.1 fLoginResultCallBack

## Login Status Callback Function

Member	Data Type	Description
IUserID	LONG	User ID, which is returned by <a href="#"><u>NET_DVR_Login_V40</u></a> .
dwResult	DWORD	Login status: 0-asynchronously logging in failed, 1-asynchronously logged in.
lpDeviceInfo	<a href="#"><u>NET_DVR_DEVICEINFO_V40</u></a>	Device information, such as serial No., channel, capability, and so on.
pUser	void*	User data.

### 3.34.2 fRemoteConfigCallback

Function for calling back the persistent connection status and data to be transmitted.

#### Callback Function Definition

```
void(CALLBACK *fRemoteConfigCallback) (
    DWORD      dwType,
    void       *lpBuffer,
    DWORD      dwBufLen,
    void       *pUserData
);
```

#### Parameters

##### dwType

[OUT] Connection statuses, see the macro definitions below:

```
enum _NET_SDK_CALLBACK_TYPE_ {
    NET_SDK_CALLBACK_TYPE_STATUS     = 0,
    NET_SDK_CALLBACK_TYPE_PROGRESS   = 1,
    NET_SDK_CALLBACK_TYPE_DATA      = 2
} NET_SDK_CALLBACK_TYPE
```

##### NET\_SDK\_CALLBACK\_TYPE\_STATUS

Connection status.

##### NET\_SDK\_CALLBACK\_TYPE\_PROGRESS

Connection progress.

##### NET\_SDK\_CALLBACK\_TYPE\_DATA

Related data to be called back.

**lpBuffer**

[OUT] Pointer of buffer for saving progress, status, and related data to be called back, which relates to **dwType**, see details in the following table.

<b>dwType</b>	<b>lpBuffer</b>
NET_SDK_CALLBACK_TYPE_STATUS	If <b>dwBufLen</b> is 4, <b>lpBuffer</b> is 4-byte connection status; if <b>dwBufLen</b> is 8, <b>lpBuffer</b> consists of 4-byte connection status and 4-byte error code. The connection status is enumerated in <a href="#"><u>NET_SDK_CALLBACK_STATUS_NORMAL</u></a> .
NET_SDK_CALLBACK_TYPE_PROGRESS	Connection progress value.
NET_SDK_CALLBACK_TYPE_DATA	Data structures to be returned, which are different according to different commands ( <b>dwCommand</b> ) in <a href="#"><u>NET_DVR_StartRemoteConfig</u></a> .

**dwBufLen**

[OUT] Buffer size.

**pUserData**

[OUT] User data.

### 3.34.3 MSGCallBack

Alarm/event information callback function.

#### Callback Function Definition

```
typedef void (CALLBACK *MSGCallBack) (
    LONG           lCommand,
    NET_DVR_ALARMER *pAlarmer,
    char            *pAlarmInfo,
    DWORD           dwBufLen,
    void            *pUser
);
```

#### Parameters

**lCommand**

[OUT] Uploaded message type. You can distinguish the alarm/event information via the type.

**pAlarmer**

[OUT] Alarm device information, including serial No., IP address, login handle, and so on, see details in **NET DVR ALARMER**.

### **pAlarmInfo**

[OUT] Alarm/event information, the details are returned in different structures according to **ICommand**.

### **dwBufLen**

[OUT] Size of alarm/event information buffer.

### **pUser**

[OUT] User data.

# Appendix A. Data Structure

## A.1 CHAR\_ENCODE\_CONVERT

Encoding type conversion callback function.

### Callback Function Definition

```
typedef int(CALLBACK *CHAR_ENCODE_CONVERT) (
    char      *pInput,
    DWORD     dwInputLen,
    DWORD     dwInEncodeType,
    char      *pOutput,
    DWORD     dwOutputLen,
    DWORD     dwOutEncodeType
);
```

### Parameters

#### pInput

[IN] Input string, whose memory and size is applied and provided by the third-party platform

#### dwInputLen

[IN] Input buffer size.

#### dwInEncodeType

[IN] Encoding types of input string: 0-no encoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift\_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western Europe).

#### pOutput

[OUT] Output string, whose memory is applied by the third-party platform.

#### dwOutputLen

[OUT] Output buffer size.

#### dwOutEncodeType

[OUT] Encoding types of output string: 0-no encoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift\_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western Europe).

### Return Values

Return *-1* for failure, and return *0* for success.

## A.2 NET\_ALARM\_RECORD\_EXCEPTION

### Structure about Recording Exception Alarm Information

Member	Data Type	Description
<b>byReason</b>	BYTE	Exception reason: 0-video volume full, 1-video volume exception, 2-no available video volume.
<b>byRes1</b>	BYTE[]	Reserved, set to 0. The maximum array length is 3 bytes.
<b>sVolumeName</b>	BYTE[]	Video volume name, the maximum array length is "MAX_VOLUMENAME_LEN" (32 bytes).
<b>dwVolumeID</b>	DWORD	Video volume ID, or HDD No.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 452 bytes.

## A.3 NET\_ALARM\_STREAM\_EXCEPTION

### Structure about Video Exception Alarm Information

Member	Data Type	Description
<b>struIP</b>	<b><i>NET_DVR_IPADDR_UN ION</i></b>	IP address of video exception channel.
<b>dwChanNo</b>	DWORD	Channel No.
<b>dwIDIndex</b>	DWORD	Encoder ID.
<b>sName</b>	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
<b>byExceptionCase</b>	BYTE	Exception reason: 0-data writing exception, 1-network exception.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 307 bytes.

## A.4 NET\_ALARM\_RESOURCE\_USAGE

## Structure about Resource Usage Alarm Information

Member	Data Type	Description
<b>byLevel</b>	BYTE	Usage alarm level: 0-normal, 1-alarm level 1, 2-alarm level 2, 3-alarm level 3.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 491 bytes.

## A.5 NET\_ALARM\_RECORDFILE\_LOSS

### Structure about Video Loss Alarm Information

Member	Data Type	Description
<b>struInspectStart</b>	<a href="#"><u>NET_DVR_TIME_EX</u></a>	Start time of video loss check.
<b>struInspectEnd</b>	<a href="#"><u>NET_DVR_TIME_EX</u></a>	End time of video loss check.
<b>struIP</b>	<a href="#"><u>NET_DVR_IPADDR_UNION</u></a>	IP address of video loss channel.
<b>dwChanNo</b>	DWORD	Channel No.
<b>dwIDIndex</b>	DWORD	Encoder ID.
<b>sName</b>	BYTE[]	Encoder name, the maximum array length is "STREAM_ID_LEN" (32 bytes).
<b>struLossStartTime</b>	<a href="#"><u>NET_DVR_TIME_EX</u></a>	Start time of video loss.
<b>struLossEndTime</b>	<a href="#"><u>NET_DVR_TIME_EX</u></a>	End time of video loss.
<b>dwLostNum</b>	DWORD	Number of lost video files, 0xffffffff-all video files are lost.
<b>byRes</b>	BYTE[]	Reserved, set to 0. The maximum array length is 240 bytes.

## A.6 NET\_ALARM\_CVR\_SUBINFO\_UNION

## Union about CVR Alarm Information

Member	Data Type	Description
byLen	BYTE[]	Union size, the maximum array length is 492 bytes.
struRecordLost	<u><a href="#">NET_ALARM_RECORD</a></u> <u><a href="#">FILE LOSS</a></u>	Video loss alarm information, the value of <b>dwAlarmType</b> in <u><a href="#">NET_DVR_ALARMINFO_DEV_V40</a></u> is 8.
struStreamException	<u><a href="#">NET_ALARM_STREAM_EXCEPTION</a></u>	Streaming exception alarm information, the value of <b>dwAlarmType</b> in <u><a href="#">NET_DVR_ALARMINFO_DEV_V40</a></u> is 9.
struResourceUsage	<u><a href="#">NET_ALARM_RESOURCE_USAGE</a></u>	Resource usage alarm information, the value of <b>dwAlarmType</b> in <u><a href="#">NET_DVR_ALARMINFO_DEV_V40</a></u> is 10.
struRecordException	<u><a href="#">NET_ALARM_RECORD_EXCEPTION</a></u>	Recording exception alarm information, the value of <b>dwAlarmType</b> in <u><a href="#">NET_DVR_ALARMINFO_DEV_V40</a></u> is 12.

## A.7 NET\_DVR\_ACS\_ALARM\_INFO

### Structure about Access Control Alarm/Event Information

Member	Data Type	Description
dwSize	DWORD	Structure size.
dwMajor	DWORD	Major alarm/event types, see details in <u><a href="#">Access Control Event Types</a></u> .
dwMinor	DWORD	Minor alarm/event types, see details in <u><a href="#">Access Control Event Types</a></u> .
struTime	<u><a href="#">NET_DVR_TIME</a></u>	Alarm time information.
sNetUser	Array [BYTE]	User name for network operation. The maximum size is 16 bytes (the value of the

Member	Data Type	Description
		macro definition "MAX_NAMELEN").
struRemoteHostAddr	<a href="#"><u>NET_DVR_IPADDR_UNION</u></a>	IP address of the remote access controller.
struAcsEventInfo	<a href="#"><u>NET_DVR_ACS_EVENT_INFO</u></a>	Access control event details.
dwPicDataLen	DWORD	Picture size, 0: no picture, non-0: picture data exist.
pPicData	char*	Picture data.
wInductiveEventType	WORD	Inductive event type, 0-invalid. The alarm event types will be distinguished according to the inductive event type if <b>wInductiveEventType</b> is not 0; otherwise, the alarm event types will be distinguished according to <b>dwMajor</b> and <b>dwMinor</b> .
byPicTransType	BYTE	Picture data transmission mode: 0-binary, 1-URL.
byRes1	BYTE	Reserved.
dwIOTChannelNo	DWORD	IOT channel No.
pAcsEventInfoExtend	char*	When <b>byAcsEventInfoExtend</b> is set to 1, it points to the structure <a href="#"><u>NET_DVR_ACS_EVENT_INFO_EXTEND</u></a> .
byAcsEventInfoExtend	BYTE	Whether <b>pAcsEventInfoExtend</b> is valid: 0-no, 1-yes.
byTimeType	BYTE	Time type: 0-device's local time, 1-UTC time (it is the same as <b>struTime</b> ).
byRes2	BYTE	Reserved.
byAcsEventInfoExtendV20	BYTE	Whether the member <b>pAcsEventInfoExtendV20</b> is

Member	Data Type	Description
		valid: 0-invalid, 1-valid. If this member is valid, the member <b>byAcsEventInfoExtend</b> must be valid.
<b>pAcsEventInfoExtendV20</b>	char*	When <b>byAcsEventInfoExtendV20</b> is set to 1, it points to the structure <b><u>NET_DVRACS_EVENT_INFO_EXTEND_V20</u></b> .
<b>byRes</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 4 bytes.

## A.8 NET\_DVR\_ACS\_CFG

Access controller configuration parameter structure.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byRS485Backup;
    BYTE       byShowCapPic;
    BYTE       byShowCardNo;
    BYTE       byShowUserInfo;
    BYTE       byOverlayUserInfo;
    BYTE       byVoicePrompt;
    BYTE       byUploadCapPic;
    BYTE       bySaveCapPic;
    BYTE       byInputCardNo;
    BYTE       byEnableWifiDetect;
    BYTE       byEnable3G4G;
    BYTE       byProtocol;
    BYTE       byRes[500];
}NET_DVR_ACS_CFG, *LPNET_DVR_ACS_CFG;
```

### Members

#### **dwSize**

Structure size

#### **byRS485Backup**

Whether to enable downstream backup function of RS485: 0-no, 1-yes

### **byShowCapPic**

Whether to display captured picture: 0-no, 1-yes

### **byShowCardNo**

Whether to display card No.: 0-no, 1-yes

### **byShowUserInfo**

Whether to display user information: 0-no, 1-yes

### **byOverlayUserInfo**

Whether to display user information on video: 0-no, 1-yes

### **byVoicePrompt**

Whether to enable voice prompt: 0-no, 1-yes

### **byUploadCapPic**

Whether to upload captured picture: 0-no, 1-yes

### **bySaveCapPic**

Whether to save captured picture: 0-no, 1-yes

### **byInputCardNo**

Whether to allow entering card No. by keypad: 0-no, 1-yes

### **byEnableWifiDetect**

Whether to enable Wi-Fi probe: 0-no, 1-yes

### **byEnable3G4G**

Whether to enable 3G/4G: 0-no, 1-yes

### **byProtocol**

Communication protocol type of the card reader: 0-private protocol (default), 1-OSDP protocol.

### **byRes**

Reserved, set to 0

## A.9 NET\_DVR\_ACS\_EVENT\_CFG

Access control event parameter structure.

### Structure Definition

```
struct{
    WORD           dwSize;
    WORD           dwMajor;
    WORD           dwMinor;
    NET_DVR_TIME struTime;
    BYTE          sNetUser[MAX_NAMELEN/*16*/];
```

```
NET_DVR_IPADDR           struRemoteHostAddr;
NET_DVR_ACS_EVENT_DETAIL struAcsEventInfo;
DWORD                   dwPicDataLen;
char                    *pPicData;
BYTE                   byTimeType;
BYTE                   byRes1;
DWORD                   dwQRCodeInfoLen;
DWORD                   dwVisibleLightDataLen;
DWORD                   dwThermalDataLen;
char                    *pQRCodeInfo;
char                    *pVisibleLightData;
char                    *pThermalData;
BYTE                   byRes[36];
}NET_DVR_ACS_EVENT_CFG, *LPNET_DVR_ACS_EVENT_CFG;
```

## Members

### **dwSize**

Structure size.

### **dwMajor**

Event major types, see details in [\*\*Access Control Event Types\*\*](#) .

### **dwMinor**

Event minor types, see details in [\*\*Access Control Event Types\*\*](#) .

### **struTime**

Time information, see [\*\*NET\\_DVR\\_TIME\*\*](#) for details.

### **sNetUser**

User name.

### **struRemoteHostAddr**

IP address of remote access controller, see [\*\*NET\\_DVR\\_IPADDR\\_UNION\*\*](#) for details.

### **struAcsEventInfo**

Access control event details, see [\*\*NET\\_DVR\\_ACS\\_EVENT\\_DETAIL\*\*](#) for details.

### **dwPicDataLen**

Picture size, non-0: picture data exists.

### **pPicData**

Picture data.

### **byTimeType**

Time type: 0-device local time (default), 1-UTC time (which is same as **struTime**).

### **byRes1**

Reserved.

### **dwQRCodeInfoLen**

Length of the QR code information. If this member is not 0, it indicates that there is QR code information data following after.

### **dwVisibleLightDataLen**

Length of the visible light picture captured by the thermal camera. If this member is not 0, it indicates that there is visible light picture data following after.

### **dwThermalDataLen**

Length of the thermal picture. If this member is not 0, it indicates that there is thermal picture data following after.

### **pQRCodeInfo**

Pointer of the QR code information.

### **pVisibleLightData**

Pointer of the visible light picture captured by the thermal camera.

### **pThermalData**

Pointer of the thermal picture.

### **byRes**

Reserved, set to 0.

## A.10 NET\_DVR\_ACS\_EVENT\_COND

Condition structure about getting access control events.

### Structure Definition

```
struct{
    DWORD          dwSize;
    DWORD          dwMajor;
    DWORD          dwMinor;
    NET_DVR_TIME  struStartTime;
    NET_DVR_TIME  struEndTime;
    BYTE           byCardNo[ACS_CARD_NO_LEN/*32*/];
    BYTE           byName[NAME_LEN/*32*/];
    BYTE           byPicEnable;
    BYTE           byTimeType;
    BYTE           byRes2[2];
    DWORD          dwBeginSerialNo;
    DWORD          dwEndSerialNo;
    DWORD          dwIOTChannelNo;
    WORD           wInductiveEventType;
    BYTE           bySearchType;
    BYTE           byEventAttribute;
    char           szMonitorID[NET_SDK_MONITOR_ID_LEN/*64*/];
    BYTE           byEmployeeNo[NET_SDK_EMPLOYEE_NO_LEN/*32*/];
```

```
    BYTE          byRes [140];  
}NET_DVR_ACS_EVENT_COND,*LPNET_DVR_ACS_EVENT_COND;
```

## Members

### **dwSize**

Structure size.

### **dwMajor**

Event major types, see details in [\*\*Access Control Event Types\*\*](#), 0-all.

### **dwMinor**

Event minor types, see details in [\*\*Access Control Event Types\*\*](#), 0-all.

### **struStartTime**

Start time, see [\*\*NET\\_DVR\\_TIME\*\*](#) for details.

### **struEndTime**

End time, see [\*\*NET\\_DVR\\_TIME\*\*](#) for details.

### **byCardNo**

Card No.

### **byName**

Cardholder name.

### **byPicEnable**

Whether contain pictures: 0-no, 1-yes. If this member is set to 0, all events that meet the requirements will be uploaded without pictures. If this member is set to 1, for all events that meet the requirements, the event information will be uploaded if there is no linkage picture, and the event information along with the linkage pictures will be uploaded if there are any.

### **byTimeType**

Time type: 0-device local time (default), 1-UTC time (which is same as **struStartTime** and **struEndTime**).

### **byRes2**

Reserved, set to 0.

### **dwBeginSerialNo**

Start serial No.: 0-all.

### **dwEndSerialNo**

End serial No.: 0-all.

### **dwIOTChannelNo**

IOT channel No., 0-invalid.

### **wInductiveEventType**

Inductive event type, 0-invalid. The alarm event types will be distinguished according to the inductive event type if **wInductiveEventType** is not 0; otherwise, the alarm event types will be distinguished according to **dwMajor** and **dwMinor**.

### **bySearchType**

Search mode: 0-reserved, 1-search by event source (the channel No. is the non-video channel No.), 2-search by monitoring resource ID.

### **byEventAttribute**

Event attribute: 0-undefined, 1-valid authentication, 2-other.

### **szMonitorID**

Monitoring resource ID which consists of device serial No., channel type, and No. For example, the access point ID is device serial No.+ "DOOR" + door No.

### **byEmployeeNo**

Employee No. (person ID)

### **byRes**

Reserved, set to 0.

## A.11 NET\_DVR\_ACS\_EVENT\_DETAIL

Access control event details structure.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byCardNo[ACS_CARD_NO_LEN/*32*/];
    BYTE       byCardType;
    BYTE       byAllowListNo;
    BYTE       byReportChannel;
    BYTE       byCardReaderKind;
    DWORD     dwCardReaderNo;
    DWORD     dwDoorNo;
    DWORD     dwVerifyNo;
    DWORD     dwAlarmInNo;
    DWORD     dwAlarmOutNo;
    DWORD     dwCaseSensorNo;
    DWORD     dwRs485No;
    DWORD     dwMultiCardGroupNo;
    WORD      wAccessChannel;
    BYTE       byDeviceNo;
    BYTE       byDistractControlNo;
    DWORD     dwEmployeeNo;
    WORD      wLocalControllerID;
    BYTE       byInternetAccess;
    BYTE       byType;
```

```

    BYTE      byMACAddr[MACADDR_LEN/*6*/];
    BYTE      bySwipeCardType;
    BYTE      byEventAttribute;
    DWORD     dwSerialNo;
    BYTE      byChannelControllerID;
    BYTE      byChannelControllerLampID;
    BYTE      byChannelControllerIRAdaptorID;
    BYTE      byChannelControllerIREmitterID;
    DWORD     dwRecordChannelNum;
    char      *pRecordChannelData;
    BYTE      byUserType;
    BYTE      byCurrentVerifyMode;
    BYTE      byAttendanceStatus;
    BYTE      byStatusValue;
    BYTE      byEmployeeNo[NET_SDK_EMPLOYEE_NO_LEN/*32*/];
    BYTE      byRes1;
    BYTE      byMask;
    BYTE      byThermometryUnit;
    BYTE      byIsAbnormalTemperature;
    float     fCurrTemperature;
    NET_VCA_POINT struRegionCoordinates;
    BYTE      byHealthCode;
    BYTE      byRes[47];
}NET_DVR_ACS_EVENT_DETAIL, *LPNET_DVR_ACS_EVENT_DETAIL;

```

## Members

### **dwSize**

Structure size.

### **byCardNo**

Card No.: 0-invalid.

### **byCardType**

Card types: 0-invalid, 1-normal card, 2-disabled card, 3-blocklist card, 4-patrol card, 5-duress card, 6-super card, 7-visitor card.

### **byAllowListNo**

Allowlist No., which is between 1 and 8, but if the value is 0, it is invalid.

### **byReportChannel**

Event uploading channel types: 0-invalid, 1-upload in arming mode, 2-upload by central group 1, 3-upload by central group 2.

### **byCardReaderKind**

Authentication device types: 0-invalid, 1-IC card reader, 2-ID card reader, 3-QR code scanner, 4-fingerprint module.

### **dwCardReaderNo**

Authentication device No.: 0-invalid.

**dwDoorNo**

Door or floor No.: 0-invalid. For Turnstile (swing barrier), door No.1 refers to entrance, and door No.2 refers to exist.

**dwVerifyNo**

Multiple authentication No.: 0-invalid

**dwAlarmInNo**

Alarm input No.: 0-invalid

**dwAlarmOutNo**

Alarm output No.: 0-invalid

**dwCaseSensorNo**

Event trigger No.

**dwRs485No**

RS485 channel No.: 0-invalid.

**dwMultiCardGroupNo**

Group No.

**wAccessChannel**

Turnstile No.

**byDeviceNo**

Device No.: 0-invalid.

**byDistractControlNo**

Distributed controller No.: 0-invalid.

**dwEmployeeNo**

Employee No.: 0-invalid.

**wLocalControllerID**

Distributed access controller No.: 0-access controller, 0 to 64: distributed access controller.

**byInternetAccess**

Network interface No.: 1-upstream network interface No.1, 2-upstream network interface No.2, 3-downstream network interface No.1.

**byType**

Zone types: 0-instant alarm zone, 1-24-hour alarm zone, 2-delayed zone, 3-internal zone, 4-key zone, 5-fire alarm zone, 6-perimeter protection, 7-24-hour silent alarm zone, 8-24-hour auxiliary zone, 9-24-hour shock alarm zone, 10-emergency door open alarm zone, 11-emergency door closed alarm zone, off-none

**byMACAddr**

Physical address, 0-invalid.

## **bySwipeCardType**

Card swiping type: 0-invalid, 1-QR code.

## **byEventAttribute**

Event attribute: 0-undefined, 1-valid authentication, 2-other.

## **dwSerialNo**

Event serial No.: 0-invalid, which is used to judge whether the event loss occurred.

## **byChannelControllerID**

Lane controller No.: 0-invalid, 1-main lane controller, 2-sub-lane controller.

## **byChannelControllerLampID**

Light board No. of lane controller, which is between 1 and 255, 0-invalid

## **byChannelControllerIRAdaptorID**

IR adaptor No. of lane controller, which is between 1 and 255, 0-invalid.

## **byChannelControllerIREmitterID**

Active infrared intrusion detector No. of lane controller, which is between 1 and 255, 0-invalid.

## **dwRecordChannelNum**

Number of recording channels.

## **pRecordChannelData**

Recording channel, the size depends on **dwRecordChannelNum**.

## **byUserType**

Person type: 0-invalid, 1-resident, 2-visitor, 3-person in blocklist, 4-administrator.

## **byCurrentVerifyMode**

Authentication mode: 0-invalid, 1-sleepy, 2-card+password, 3-card, 4-card or password, 5-fingerprint, 6-fingerprint+password, 7-fingerprint or card, 8-fingerprint+card, 9-fingerprint+card +password, 10-face or fingerprint or card or password, 11-face+fingerprint, 12-face+password, 13-face+card, 14-face, 15-employee No.+password, 16-fingerprint or password, 17-employee No.+fingerprint, 18-employee No.+fingerprint+password, 19-face+fingerprint+card, 20-face +password+fingerprint, 21-employee No.+face, 22-face or face+card, 23-fingerprint or face, 24-card or face or password, 25-card or face, 26-card or face or fingerprint, 27-card or fingerprint or password.

## **byAttendanceStatus**

Attendance status: 0-undefined, 1-check in, 2-check out, 3-break out, 4-break in, 5-overtime in, 6-overtime out.

## **byStatusValue**

Attendance status value.

## **byEmployeeNo**

Employee No. (person ID). Both **byEmployeeNo** and **dwEmployeeNo** should be transferred by the device. The **byEmployeeNo** will be parsed by the upper-level platform or client first. If the **byEmployeeNo** is NULL, the **dwEmployeeNo** will be parsed.

### **byRes1**

Reserved.

### **byMask**

Whether the person is wearing mask or not: 0-reserved, 1-unknown, 2-not wearing mask, 3-wearing mask.

### **byThermometryUnit**

Temperature unit: 0-Celsius (default), 1-Fahrenheit, 2-Kelvin.

### **byIsAbnormalTemperature**

Whether the face temperature is abnormal: 1-yes, 0-no.

### **fCurrTemperature**

Face temperature which is accurate to one decimal place.

### **struRegionCoordinates**

Face temperature's coordinates, see details in the structure [NET\\_VCA\\_POINT](#).

### **byHealthCode**

Health code status: 0 (no request), 1 (no health code), 2 (green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6 (other error, e.g., searching failed due to API exception), 7 (searching for the health code timed out).

### **byRes**

Reserved, set to 0.

## **A.12 NET\_DVR\_ACS\_EVENT\_INFO**

### **Structure about Extended Access Control Event Details**

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>byCardNo</b>	Array [BYTE]	Card No., 0-invalid. Some special cards' numbers are listed as the follows: "18446744073709551613"-supper card, "18446744073709551614"-duress card,

Member	Data Type	Description
		"18446744073709551615"- invalid card. The maximum size is 32 bytes (the value of the macro definition "ACS_CARD_NO_LEN").
<b>byCardType</b>	BYTE	Card types: 0-invalid, 1-normal card, 2-disability card, 3-blocklist card, 4-patrol card, 5-duress card, 6-super card, 7-visitor card, 8-dismiss card.
<b>byAllowListNo</b>	BYTE	Allowlist No., which is between 1 and 8, but if the value is 0, it is invalid.
<b>byReportChannel</b>	BYTE	Event uploading channel types: 0-invalid, 1-upload in arming mode, 2-upload by central group 1, 3-upload by central group 2.
<b>byCardReaderKind</b>	BYTE	Authentication device types: 0-invalid, 1-IC card reader, 2-ID card reader, 3-QR code scanner, 4-fingerprint module.
<b>dwCardReaderNo</b>	DWORD	Authentication device No.: 0-invalid.
<b>dwDoorNo</b>	DWORD	Door or floor No.: 0-invalid. For turnstile (swing barrier), door No.1 refers to entrance, and door No.2 refers to exit.
<b>dwVerifyNo</b>	DWORD	Multiple authentication No.: 0-invalid.
<b>dwAlarmInNo</b>	DWORD	Alarm input No.: 0-invalid.
<b>dwAlarmOutNo</b>	DWORD	Alarm output No.: 0-invalid.
<b>dwCaseSensorNo</b>	DWORD	Event trigger No.
<b>dwRs485No</b>	DWORD	RS-485 channel No.: 0-invalid.
<b>dwMultiCardGroupNo</b>	DWORD	Group No.

Member	Data Type	Description
wAccessChannel	WORD	Turnstile No.
byDeviceNo	BYTE	Device No.: 0-invalid.
byDistractControlNo	BYTE	Distributed controller No.: 0-invalid.
dwEmployeeNo	DWORD	Employee ID: 0-invalid.
wLocalControllerID	WORD	Distributed access controller No.: 0-access controller, 0 to 64: distributed access controller.
byInternetAccess	BYTE	Network interface No.: 1-upstream network interface No.1, 2-upstream network interface No.2, 3-downstream network interface No.1.
byType	BYTE	Zone types: 0-instant zone, 1-24-hour zone, 2-delayed zone, 3-internal zone, 4-key zone, 5-fire alarm zone, 6-perimeter zone, 7-24-hour silent zone, 8-24-hour auxiliary zone, 9-24-hour shock zone, 10-emergency door open alarm zone, 11-emergency door closed alarm zone, 0xff-none.
byMACAddr	Array [BYTE]	Physical address, 0-invalid. The maximum size is 6 bytes (the value of the macro definition "MACADDR_LEN").
bySwipeCardType	BYTE	Card swiping type: 0-invalid, 1-QR code.
byMask	BYTE	Whether the person is wearing mask: 0-reserved, 1-unknown, 2-not wearing mask, 3-wearing mask.

Member	Data Type	Description
<b>dwSerialNo</b>	DWORD	Event serial No.: 0-invalid, which is used to check whether the event loss occurred.
<b>byChannelControllerID</b>	BYTE	Lane controller No.: 0-invalid, 1-main lane controller, 2-sub-lane controller.
<b>byChannelControllerLampID</b>	BYTE	Light board No. of the lane controller, which is between 1 and 255, 0-invalid.
<b>byChannelControllerIRAdaptorID</b>	BYTE	IR adaptor No. of the lane controller, which is between 1 and 255, 0-invalid.
<b>byChannelControllerIREmitterID</b>	BYTE	Active infrared intrusion detector No. of the lane controller, which is between 1 and 255, 0-invalid.
<b>byHelmet</b>	BYTE	Whether the person is wearing hard hat: 1-unknown, 2-no, 3-yes.
<b>byHealthCode</b>	BYTE	Health code status: 0 (no request), 1 (no health code), 2 (green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6 (other error, e.g., searching failed due to API exception), 7 (searching for the health code timed out).
<b>byRes</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 2 bytes.

## A.13 NET\_DVR\_ACS\_EVENT\_INFO\_EXTEND

## Structure about Extended Access Control Event Information

Member	Data Type	Description
dwFrontSerialNo	DWORD	Event serial No., 0-invalid. If this member is set to 0, the platform will check whether the event is lost by <b>dwSerialNo</b> ; otherwise, the platform will check whether the event is lost by both <b>dwFrontSerialNo</b> and <b>dwSerialNo</b> . This member is used for discontinuous <b>dwSerialNo</b> after alarm subscription.
byUserType	BYTE	Person type: 0-invalid, 1-normal person (resident), 2-visitor, 3-person in the blocklist, 4-administrator.
byCurrentVerifyMode	BYTE	Current authentication mode of the card reader: 0-invalid, 1-sleepy, 2-card+password, 3-card, 4-card or password, 5-fingerprint, 6-fingerprint+password, 7-fingerprint or card, 8-fingerprint+card, 9-fingerprint+card+password, 10-face or fingerprint or card or password, 11-face+fingerprint, 12-face+password, 13-face+card, 14-face, 15-employee No.+password, 16-fingerprint or password, 17-employee No.+fingerprint, 18-employee No.+fingerprint+password, 19-face+fingerprint+card, 20-face+password+fingerprint, 21-employee No.+face, 22-face or face+card, 23-fingerprint or face, 24-card or face or password, 25-card or face, 26-card or face or fingerprint, 27-card or fingerprint or password.
byCurrentEvent	BYTE	Whether it is a real-time event: 0-invalid, 1-yes (real-time event), 2-no (offline event).
byPurePwdVerifyEnable	BYTE	Whether the device supports opening the door only by password: 1-yes, 0-no. For opening the door only by password: 1. The password in "XXX or password" in the authentication mode refers to the person's password (the value of the node <b>password</b> in JSON_UserInfo); 2. The device will not check the duplication of the password, and the upper

Member	Data Type	Description
		platform should ensure that the password is unique; 3. The password cannot be added, deleted, edited, or searched for on the device locally.
byEmployeeNo	BYTE[]	Employee No. (person ID). Both <b>byEmployeeNo</b> and <b>dwEmployeeNo</b> should be transferred by the device. The <b>byEmployeeNo</b> will be parsed by the upper-layer platform or client first. If the <b>byEmployeeNo</b> is not configured, the <b>dwEmployeeNo</b> will be parsed. The maximum length is "NET_SDK_EMPLOYEE_NO_LEN" (32 bytes).
byAttendanceStatus	BYTE	Attendance status: 0-undefined, 1-check in, 2-check out, 3-break out, 4-break in, 5-overtime in, 6-overtime out.
byStatusValue	BYTE	Attendance status value.
byRes2	BYTE[]	Reserved. The maximum length is 2 bytes.
byUUID	BYTE[]	UUID, this member is only used when accessing EZVIZ platform. The maximum length is "NET_SDK_UUID_LEN" (36 bytes).
byDeviceName	BYTE[]	Device serial No. The maximum length is "NET_DEV_NAME_LEN" (64 bytes).
dwBodyTemp	DWORD	Skin-surface temperature, which equals to actual temperature value (a float number) × 1000.
byMaskEnabled	BYTE	Whether the person is wearing mask: 1 (yes), 2 (no).
byRes	BYTE[]	Reserved. The maximum length is 19 bytes.

## See Also

[NET\\_DVR\\_ACS\\_ALARM\\_INFO](#)

## A.14 NET\_DVR\_ACS\_EVENT\_INFO\_EXTEND\_V20

## Structure about Extended Access Control Event Information (V20)

Member	Data Type	Description
<b>byRemoteCheck</b>	BYTE	Whether remote verification is required: 0-invalid, 1-no (default), 2-yes.
<b>byThermometryUnit</b>	BYTE	Temperature unit: 0-Celsius (default), 1-Fahrenheit, 3-Kelvin.
<b>byIsAbnormalTemperature</b>	BYTE	Whether the face temperature is abnormal: 1-yes, 0-no.
<b>byRes2</b>	BYTE	Reserved.
<b>fCurrTemperature</b>	float	Face temperature, it is accurate to one decimal place.
<b>struRegionCoordinates</b>	<u>NET_VCA_POINT</u>	Face temperature's coordinates.
<b>dwQRCodeInfoLen</b>	DWORD	Data size of the QR code information. If this member is not 0, it indicates that the QR code information data exists.
<b>dwVisibleLightDataLen</b>	DWORD	Data size of the visible light picture captured by the thermal camera. If this member is not 0, it indicates that the visible light picture data exists.
<b>dwThermalDataLen</b>	DWORD	Data size of the thermal picture. If this member is not 0, it indicates that the thermal picture data exists.
<b>pQRCodeInfo</b>	char*	Pointer of the QR code information.
<b>pVisibleLightData</b>	char*	Pointer of the visible light picture captured by the thermal camera.
<b>pThermalData</b>	char*	Pointer of the thermal picture.

Member	Data Type	Description
<b>byAttendanceLabel</b>	Array [BYTE]	Custom attendance name. The maximum size is 64 bytes.
<b>byRes</b>	Array [BYTE]	Reserved. The maximum size is 960 bytes.

## A.15 NET\_DVR\_ACS\_WORK\_STATUS\_V50

Access controller working status structure.

### Structure Definition

```
struct{
    WORD      dwSize;
    BYTE     byDoorLockStatus[MAX_DOOR_NUM/*256*/];
    BYTE     byDoorStatus[MAX_DOOR_NUM/*256*/];
    BYTE     byMagneticStatus[MAX_DOOR_NUM/*256*/];
    BYTE     byCaseStatus[MAX_CASE_SENSOR_NUM/*8*/];
    WORD     wBatteryVoltage;
    BYTE     byBatteryLowVoltage;
    BYTE     byPowerSupplyStatus;
    BYTE     byMultiDoorInterlockStatus;
    BYTE     byAntiSneakStatus;
    BYTE     byHostAntiDismantleStatus;
    BYTE     byIndicatorLightStatus;
    BYTE     byCardReaderOnlineStatus[MAX_CARD_READER_NUM/*512*/];
    BYTE     byCardReaderAntiDismantleStatus[MAX_CARD_READER_NUM/*512*/];
    BYTE     byCardReaderVerifyMode[MAX_CARD_READER_NUM/*512*/];
    BYTE     bySetupAlarmStatus[MAX_ALARMHOST_ALARMIN_NUM/*512*/];
    BYTE     byAlarmInStatus[MAX_ALARMHOST_ALARMIN_NUM/*512*/];
    BYTE     byAlarmOutStatus[MAX_ALARMHOST_ALARMOUT_NUM/*512*/];
    DWORD    dwCardNum;
    BYTE     byFireAlarmStatus;
    BYTE     byBatteryChargeStatus;
    BYTE     byMasterChannelControllerStatus;
    BYTE     bySlaveChannelControllerStatus;
    BYTE     byAntiSneakServerStatus;
    BYTE     byRes3[3];
    DWORD    dwAllowFaceNum;
    DWORD    dwBlockFaceNum;
    BYTE     byRes2[108];
}NET_DVR_ACS_WORK_STATUS_V50,*LPNET_DVR_ACS_WORK_STATUS_V50;
```

### Members

#### **dwSize**

Structure size

## **byDoorLockStatus**

Lock status (or elevator relay status), 0-closed, 1-open, 2-short circuit alarm, 3-open circuit alarm, 4-exception alarm

## **byDoorStatus**

Door status (or elevator status): 1-sleepy, 2-open (for elevator: free status), 3-closed (for elevator: disabled status), 4-normal (for elevator: controlled status).

## **byMagneticStatus**

Magnet status: 0-closed, 1-open, 2-short circuit alarm, 3-open circuit alarm, 4-exception alarm.

## **byCaseStatus**

Alarm input status: 0-no input, 1-with input.

## **wBatteryVoltage**

Storage battery voltage, the actual value equals to the 10 multiples of **wBatteryVoltage**, unit: volt.

## **byBatteryLowVoltage**

Whether the storage battery is in low voltage status: 0-no, 1-yes.

## **byPowerSupplyStatus**

Device power supply status: 1-AC, 2-storage battery.

## **byMultiDoorInterlockStatus**

Multi-door interlocking status: 0-disabled, 1-enabled.

## **byAntiSneakStatus**

Anti-passing back status: 0-disabled, 1-enabled.

## **byHostAntiDismantleStatus**

Controller tampering status: 0-disabled, 1-enabled.

## **byIndicatorLightStatus**

Indicator status: 0-offline, 1-online.

## **byCardReaderOnlineStatus**

Fingerprint and card reader status: 0-offline, 1-online.

## **byCardReaderAntiDismantleStatus**

Fingerprint and card reader tampering status: 0-offline, 1-online.

## **byCardReaderVerifyMode**

Authentication types: 0-invalid, 1-sleepy, 2-card+password, 3-card, 4-card or password, 5-fingerprint, 6-fingerprint+password, 7-fingerprint or card, 8-fingerprint+card, 9-fingerprint+card +password, 10-face+fingerprint+card+password, 11-face+fingerprint, 12-face+password, 13-face +card, 14-face, 15-employee ID+password, 16-fingerprint or password, 17-employee ID

+fingerprint, 18-employee ID+fingerprint+password, 19-face+fingerprint+card, 20-face+fingerprint+password, 21-employee ID+face, 22-face/face+card, 23-fingerprint/face, 24-card/face/password.

### **bySetupAlarmStatus**

Alarm input arming status: 0-disarmed, 1-armed

### **byAlarmInStatus**

Alarm input status: 0-no alarm, 1-in alarm.

### **byAlarmOutStatus**

Alarm output status: 0-no alarm, 1-in alarm.

### **dwCardNum**

Number of added cards.

### **byFireAlarmStatus**

Fire alarm status: 0-normal, 1-short circuit alarm, 2-open circuit alarm.

### **byBatteryChargeStatus**

Battery charging status: 0-invalid, 1-charging, 2-unchanged.

### **byMasterChannelControllerStatus**

Online status of main lane controller online status: 0-invalid, 1-offline, 2-online.

### **bySlaveChannelControllerStatus**

Online status of sub-lane controller online status: 0-invalid, 1-offline, 2-online.

### **byAntiSneakServerStatus**

Anti-passing back server status: 0-invalid, 1-disabled, 2-normal, 3-disconnected.

### **byRes3**

Reserved, set to 0.

### **dwAllowFaceNum**

The number of face pictures in allowlist.

### **wBlockFaceNum**

The number of face pictures in blocklist.

### **byRes2**

Reserved, set to 0

## **A.16 NET\_DVR\_ALARMER**

## Alarm Device Information Structure

Member	Data Type	Description
byUserIDValid	BYTE	Whether the user ID is valid: 0-no, 1-yes
bySerialValid	BYTE	Whether the serial No. is valid: 0-no, 1-yes
byVersionValid	BYTE	Whether the version No. is valid: 0-no, 1-yes
byDeviceNameValid	BYTE	Whether the device name is valid: 0-no, 1-yes
byMacAddrValid	BYTE	Whether the MAC address is valid: 0-no, 1-yes
byLinkPortValid	BYTE	Whether the login port No. is valid: 0-no, 1-yes
byDeviceIPValid	BYTE	Whether the device IP address is valid: 0-no, 1-yes
bySocketIPValid	BYTE	Whether the Socket IP address is valid: 0-no, 1-yes
lUserID	LONG	Value returned by <a href="#"><u>NET_DVR_Login_V40</u></a> , it is valid when arming.
sSerialNumber	Array of BYTE	Serial No.
dwDeviceVersion	DWORD	Version information
sDeviceName	Array of char	Device name
byMacAddr	Array of BYTE	MAC address
wLinkPort	WORD	Device communication port No.
sDeviceIP	Array of char	Device IP address
sSocketIP	Array of char	Socket IP address when actively uploading alarm.
byIpProtocol	BYTE	Network protocol: 0-IPv4, 1-IPv6
byRes2	Array of BYTE	Reserved, set to 0.

## A.17 NET\_DVR\_ALARMINFO\_DEV

## Device Alarm Information Structure

Memeber	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel.
<b>struTime</b>		Alarm time
<b>byRes</b>	Array of BYTE	Reserved, set to 0.
<b>dwNumber</b>	DWORD	Number of alarm triggered channels.
<b>pNO</b>	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

### Remarks

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

## A.18 NET\_DVR\_ALARMINFO\_DEV\_V40

### Structure about CVR Alarm Information

Member	Data Type	Description
<b>dwAlarmType</b>	DWORD	Alarm categories: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm,

Member	Data Type	Description
		11-CVR exception recovered, 12-recording exception.
struTime	<u>NET_DVR_TIME</u>	Alarm time
uSubAlarmInfo	<u>NET_ALARM_CVR_SUBINFO_UNION</u>	CVR alarm information structure, and it is valid when the alarm type is 8, 9, 10, and 12.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 256 bytes.
dwNumber	DWORD	Number of alarm triggered channels.
pNO	WORD*	Channel No. or disk No., which ranges from 0 to 65535.

### Remarks

For **pNO**: if **dwAlarmType** is 0, 3, 6, or 7, it may be channel No.; if **dwAlarmType** is 5, it may be disk No.

## A.19 NET\_DVR\_ALARMINFO\_V30

### Structure About Uploaded Alarm Information

Member	Data Type	Description
dwAlarmType	DWORD	Alarm types: 0-alarm input alarm of encoder, 1-second private volume damaged, 2-NVR disconnected, 3-encoder exception, 4-system clock exception, 5-the remaining capacity of the recording volume is too low, 6-motion detection alarm of encoder or encoding channel, 7-video tampering alarm of encoder or encoding channel, 8-video loss alarm, 9-real-time health monitoring alarm, 10-usage alarm, 11-CVR exception recovered, 12-recording exception.
dwAlarmInputNumber	DWORD	Alarm input No., it is valid when alarm type is 0 or 23
byAlarmOutputNumber	Array of BYTE	The triggered alarm output No. E.g. dwAlarmOutputNumber[0]==1 indicates that

Member	Data Type	Description
		alarm output No.1 is triggered; dwAlarmOutputNumber[1]==1 indicates that alarm output No.2 is triggered.
byAlarmRelateChannel	Array of BYTE	The triggered recording channel No.: 0-not triggered, 1-triggered. E.g. dwAlarmRelateChannel[0]==1 indicates that the channel No.1 is triggered to record.
byChannel	Array of BYTE	Alarm channel, it is valid when alarm type is 2, 3, 6, 9, 10 or 11. E.g. dwChannel[0]==1 indicates that the channel No. is in alarm.
byDiskNumber	Array of BYTE	Alarm HDD, it is valid when alarm type is 1, 4, or 5. E.g. dwDiskNumber [0]==1 indicates that the HDD No.1 is abnormal.

### Remarks

The time interval to upload the alarm of face picture library changed is 1 hour; for other alarm type, the alarm information is uploaded in real-time, and the time interval is 1s. Currently, editing the time interval is not supported.

## A.20 NET\_DVR\_ALARMINFO\_V40

### Structure About Uploaded Alarm Information

Member	Data Type	Description
struAlarmFixedHeader	<a href="#"><u>NET_DVR_ALRAM_FIXED_HEADER</u></a>	Constant content in alarm information, see details in the structure .
pAlarmData	DWORD*	Variable content in alarm information

### Remarks

- The time interval to upload the alarm of face picture library changed is 1 hour; for other alarm type, the alarm information is uploaded in real-time, and the time interval is 1s. Currently, editing the time interval is not supported.
- The content of **pAlarmData** varies with the value of **dwAlarmType** in the structure [NET\\_DVR\\_ALRAM\\_FIXED\\_HEADER](#), see details in the table below:

**Table A-1 Relations Between pAlarmData and dwAlarmType**

<b>dwAlarmType</b>	<b>Description</b>	<b>pAlarmData</b>
0, 23	Alarm input alarm, pulse alarm	dwTrigerAlarmOutNum*(DWOR D) Alarm output No., +dwTrigerRecordChanNum*(WORD) Channel No.
2, 3, 6, 9, 10, 11, 13, 15, 16, 19	Video loss, motion detection, video tampering alarm, video exception, recording exception, scene change, resolution mismatched, VCA detection, PoE power supply exception, audio loss	dwAlarmChanNum*(DWORD) channel No.
1, 4, 5	HDD full, HDD uninitialized, writing to HDD failed	dwAlarmHardDiskNum*(DWOR D) HDD No.
7, 8, 12, 17, 18, 24, 25, 26	Standard mismatches, invalid login, array exception, education sharing system alarm, two-way audio request alarm, face library HDD exception, face library changed, picture changed in face picture library	None

## A.21 NET\_DVR\_ALRAM\_FIXED\_HEADER

### Structure About Constant Alarm Information

<b>Member</b>	<b>Data Type</b>	<b>Description</b>
dwAlarmType	DWORD	Alarm information type: 0-alarm input alarm, 1-HDD full, 2-video loss, 3-motion detection, 4-HDD unformatted, 5-writing to HDD failed, 6-video tampering alarm, 7-standard mismatched, 8-invalid login, 9-video exception, 10-recording exception, 11-scene change, 12-RAID exception, 13-resolution mismatched, 15-VCA detection, 16- PoE power supply exception, 17-education sharing system alarm, 18-two-way audio request alarm, 23-pulse alarm, 24-face picture

Member	Data Type	Description
		library HDD exception, 25-face picture library changed, 26-picture of face picture library changed, 27-POC exception, 28-camera FOV exception, 30-no SD card, 31-supply voltage exception, 32-PTZ locked
struAlarmTime	<a href="#"><u>NET_DVR_TIME_EX</u></a>	Alarm time
uStruAlarm	Union ( <a href="#"><u>Table 4-2</u></a> )	Alarm information union
pRes	DWORD*	Reserved.
byTimeDiffFlag	BYTE	Whether the time difference parameter is valid: 0-invalid, 1-valid.
cTimeDifferenceH	char	Time difference between time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".
cTimeDifferenceM	char	Time difference between time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 5 bytes.

**Table A-2 Union about Alarm Information Structures (uStruAlarm)**

Member	Data Type	Description
byUnionLen	Array of BYTE	Union size, which is 116 bytes.
struIOAlarm	Struct ( <a href="#"><u>Table 4-3</u></a> )	Structure about alarm input parameters
struAlarmChannel	Struct ( <a href="#"><u>Table 4-4</u></a> )	Structure about alarm channel parameters
struAlarmHardDisk	Struct ( <a href="#"><u>Table 4-5</u></a> )	Structure about HDD alarm parameters
struRecordingHost	Struct ( <a href="#"><u>Table 4-6</u></a> )	Structure about alarm parameters of education sharing system
struVoltageInstable	Struct ( <a href="#"><u>Table 4-7</u></a> )	Structure about alarm parameters of supply voltage exception
struPTLocking	Struct ( <a href="#"><u>Table 4-8</u></a> )	Structure about parameters of PTZ locked alarm

**Table A-3 Structure about Alarm Input Parameters (struIOAlarm)**

<b>Member</b>	<b>Data Type</b>	<b>Description</b>
dwAlarmInputNo	DWORD	Alarm input No.
dwTrigerAlarmOutNum	DWORD	The number of triggered alarm outputs. It is used for calculating the number of all triggered alarm outputs by <b>pAlarmData</b> in <b><i>NET_DVR_ALARMINFO_V40</i></b> , each alarm output is represented by 4 bytes.
dwTrigerRecordChanNum	DWORD	The number of triggered recording channels. It is used for calculating the number of all triggered recording channels by <b>pAlarmData</b> of <b><i>NET_DVR_ALARMINFO_V40</i></b> , each channel is represented by 4 bytes.

**Table A-4 Structure about Alarm Channel Parameters (struAlarmChannel)**

<b>Member</b>	<b>Data Type</b>	<b>Description</b>
dwAlarmChanNum	DWORD	The number of alarm channels. It is used for calculating the number of all alarm channels by <b>pAlarmData</b> of <b><i>NET_DVR_ALARMINFO_V40</i></b> , each alarm channel is represented by 4 bytes.
dwPicLen	DWORD	Size of JPEG picture.
byPicURL	BYTE	Picture data format: 0-binary data, 1-URL.
byTarget	BYTE	Detection target type: 0-not supported, 1-person, 2-vehicle.
byRes1	Array of BYTE	Reserved, the maximum size is 2 bytes.
pDataBuff	char*	Alarm picture data or URL. The pointer size is 8 bytes.
byRes3	Array of BYTE	Reserved, the maximum size is 4 bytes. This member is only available for 64-bit Windows operating system and 64-bit Linux operating system.

**Table A-5 Structure about HDD Alarm Parameters (struAlarmHardDisk)**

Member	Data Type	Description
dwAlarmHardDiskNum	DWORD	The number of alarm HDD. It is used for calculating the number of all alarm HDDs by <b>pAlarmData</b> of <b>NET_DVR_ALARMINFO_V40</b> , each alarm HDD is represented by 4 bytes.

**Table A-6 Structure about Alarm Parameters of Education Sharing System (struRecordingHost)**

Member	Data Type	Description
bySubAlarmType	BYTE	Alarm minor type: 1-one-touch post-record
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.
struRecordEndTime	<b>NET_DVR_TIME_EX</b>	Recording end time.

**Table A-7 Structure about Alarm Parameters of Supply Voltage Exception (struVoltageInstable)**

Member	Data Type	Description
fVoltageValue	float	Supply voltage, unit: V, corrects to one decimal place.
byVoltageAlarmType	BYTE	Supply voltage exception type: 0-high supply voltage, 1-low supply voltage
byRes1	Array of BYTE	Reserved, set to 0. The maximum size is 3 bytes.

**Table A-8 Structure about Parameters of PTZ Locked Alarm (struPTLocking)**

Member	Data Type	Description
fTemperature	float	Sensor temperature, which is accurate to one decimal place.
dwCustomInfoLength	DWORD	Custom information length.
pCustomInfo	BYTE*	Custom information.
byType	BYTE	PTZ locked direction: 1-panning is locked, 2-tilting is locked.
byDeicingEnabled	BYTE	Whether to enable heat for PTZ: 0-no, 1-yes.

## Remarks

**dwAlarmType==0**, 23 corresponds to the structure struIOAlarm; **dwAlarmType==2/3/6/9/10/11/13/15/16/28** corresponds to the structure struAlarmChannel; **dwAlarmType==**

1/4/5 corresponds to the structure struAlarmHardDisk; **dwAlarmType== 17** corresponds to the structure struRecordingHost; **dwAlarmType== 31** corresponds to the structure struVoltageInstable; for other value, the union is not available.

## A.22 NET\_DVR\_ALARM\_ISAPI\_INFO

### Structure about Alarm Information Transmitted Based on Text Protocol

Member	Data Type	Description
<b>pAlarmData</b>	char*	Alarm information based on text protocol (XML or JSON message without binary data).
<b>dwAlarmDataLen</b>	DWORD	Alarm data length.
<b>byDataType</b>	BYTE	Alarm data type: 0-invalid, 1-XML, 2-JSON.
<b>byPicturesNumber</b>	BYTE	The number of pictures (number of <b>pPicPackData</b> returned). When this member is 1, only one structure of <b>NET_DVR_ALARM_ISAPI_PICD ATA</b> will be returned by <b>pPicPackData</b> . When this member is larger than 1, multiple structures of <b>NET_DVR_ALARM_ISAPI_PICD ATA</b> will be returned by <b>pPicPackData</b> .
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 2 bytes.
<b>pPicPackData</b>	void*	Alarm picture structure, see <b>NET_DVR_ALARM_ISAPI_PICD ATA</b> for details.
<b>byRes</b>	Array of BYTE	Reserved. The maximum size is 32 bytes.

### Remarks

When enabling the listening mode, you should call the network configuration API based on text protocol to set the IP address for the listening service.

## A.23 NET\_DVR\_ALARM\_ISAPI\_PICDATA

### Structure about Alarm Picture Data Transmitted Based on Text Protocol

Member	Data Type	Description
<b>dwPicLen</b>	DWORD	Alarm picture data length.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 4 bytes.
<b>szFilename</b>	Array of char	Picture file saving path, including file name. The maximum size is 256 bytes.
<b>pPicData</b>	BYTE*	Pointer that pointing to the uploaded image data.

## A.24 NET\_DVR\_CAPTURE\_FACE\_CFG

Collected face data structure

### Structure Definition

```
struct{
    WORD dwSize;
    WORD dwFaceTemplate1Size;
    char *pFaceTemplate1Buffer;
    WORD dwFaceTemplate2Size;
    char *pFaceTemplate2Buffer;
    WORD dwFacePicSize;
    char *pFacePicBuffer;
    BYTE byFaceQuality1;
    BYTE byFaceQuality2;
    BYTE byCaptureProgress;
    BYTE byFacePicQuality;
    WORD dwInfraredFacePicSize;
    char *pInfraredFacePicBuffer;
    BYTE byInfraredFacePicQuality;
    BYTE byRes1[3];
    NET_DVR_FACE_FEATURE struFeature;
    BYTE byRes[56];
}NET_DVR_CAPTURE_FACE_CFG, *LPNET_DVR_CAPTURE_FACE_CFG;
```

### Members

#### **dwSize**

Structure size.

#### **dwFaceTemplate1Size**

Size of face data template 1. When its value is 0, it indicates that there is no data template 1.

#### **pFaceTemplate1Buffer**

Buffer to save face data template 1, the buffer size should be smaller than or equal to 2.5 KB.

#### **dwFaceTemplate2Size**

Size of face data template 2. When its value is 0, it indicates that there is no data template 2.

#### **pFaceTemplate2Buffer**

Buffer to save face data template 2, the buffer size should be smaller than or equal to 2.5 KB.

#### **dwFacePicSize**

Size of face picture data. When its value is 0, it indicates that there is no face picture data.

#### **pFacePicBuffer**

Buffer to save face picture data.

#### **byFaceQuality1**

Face picture quality, it is between 1 and 100.

#### **byFaceQuality2**

Face picture quality, it is between 1 and 100.

#### **byCaptureProgress**

Collection progress: 0-no face data collected, 1-collected. The face information can be parsed only when the progress value is 100.

#### **byFacePicQuality**

Face quality in the face picture.

#### **dwInfraredFacePicSize**

Size of infrared face picture data. When its value is 0, it indicates that there is no face picture data.

#### **pInfraredFacePicBuffer**

Buffer to save infrared face picture data.

#### **byInfraredFacePicQuality**

Face quality in the infrared face picture.

#### **byRes1**

Reserved.

#### **struFeature**

Feature information in the matted face picture, see details in the structure  
**NET\_DVR\_FACE\_FEATURE**.

### **byRes**

Reserved.

## A.25 NET\_DVR\_CAPTURE\_FACE\_COND

Condition structure for collecting face data.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byRes[128];
}NET_DVR_CAPTURE_FACE_COND,*LPNET_DVR_CAPTURE_FACE_COND;
```

### Members

#### **dwSize**

Structure size.

#### **byRes**

Reserved.

## A.26 NET\_DVR\_CAPTURE\_FINGERPRINT\_CFG

Fingerprint collection result structure

### Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwFingerPrintDataSize;
    BYTE       byFingerData[MAX_FINGER_PRINT_LEN/*768*/];
    DWORD      dwFingerPrintPicSize;
    char       *pFingerPrintPicBuffer;
    BYTE       byFingerNo;
    BYTE       byFingerPrintQuality;
    BYTE       byRes[62];
}NET_DVR_CAPTURE_FINGERPRINT_CFG, *LPNET_DVR_CAPTURE_FINGERPRINT_CFG;
```

### Members

#### **dwSize**

Structure size.

**dwFingerPrintDataSize**

Fingerprint data size.

**byFingerData**

Fingerprint details.

**dwFingerPrintPicSize**

Fingerprint picture size, 0-no fingerprint picture.

**pFingerPrintPicBuffer**

Buffer for saving fingerprint picture data.

**byFingerNo**

Finger No., which is between 1 and 10.

**byFingerPrintQuality**

Fingerprint quality, which is between 1 and 100.

**byRes**

Reserved, set to 0.

## A.27 NET\_DVR\_CAPTURE\_FINGERPRINT\_COND

Fingerprint collection condition structure

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byFingerPrintPicType;
    BYTE       byFingerNo;
    BYTE       byRes[126];
}NET_DVR_CAPTURE_FINGERPRINT_COND, *LPNET_DVR_CAPTURE_FINGERPRINT_COND;
```

### Members

**dwSize**

Structure size.

**byFingerPrintPicType**

Fingerprint picture type: 0-reserved.

**byFingerNo**

Finger No., which is between 1 and 10.

**byRes**

Reserved, set to 0.

## A.28 NET\_DVR\_CARD\_READER\_CFG\_V50

Fingerprint and card reader parameters structure.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byEnable;
    BYTE       byCardReaderType;
    BYTE       byOkLedPolarity;
    BYTE       byErrorLedPolarity;
    BYTE       byBuzzerPolarity;
    BYTE       bySwipeInterval;
    BYTE       byPressTimeout;
    BYTE       byEnableFailAlarm;
    BYTE       byMaxReadCardFailNum;
    BYTE       byEnableTamperCheck;
    BYTE       byOfflineCheckTime;
    BYTE       byFingerPrintCheckLevel;
    BYTE       byUseLocalController;
    BYTE       byRes1;
    WORD       wLocalControllerID;
    WORD       wLocalControllerReaderID;
    WORD       wCardReaderChannel;
    BYTE       byFingerPrintImageQuality;
    BYTE       byFingerPrintContrastTimeOut;
    BYTE       byFingerPrintRecognizeInterval;
    BYTE       byFingerPrintMatchFastMode;
    BYTE       byFingerPrintModuleSensitive;
    BYTE       byFingerPrintModuleLightCondition;
    BYTE       byFaceMatchThresholdN;
    BYTE       byFaceQuality;
    BYTE       byFaceRecognizeTimeOut;
    BYTE       byFaceRecognizeInterval;
    WORD       wCardReaderFunction;
    BYTE       byCardReaderDescription [CARD_READER_DESCRIPTION/*32*/];
    WORD       wFaceImageSensitometry;
    BYTE       byLivingBodyDetect;
    BYTE       byFaceMatchThreshold1;
    WORD       wBuzzerTime;
    BYTE       byFaceMatch1SecurityLevel;
    BYTE       byFaceMatchNSecurityLevel;
    BYTE       byEnvirMode;
    BYTE       byLiveDetLevelSet;
    BYTE       byLiveDetAntiAttackCntLimit;
    BYTE       byEnableLiveDetAntiAttack;
    BYTE       bySupportDelFPByID;
    BYTE       byRes1;
    BYTE       byFaceContrastMotionDetLevel;
```

```
    BYTE    byDayFaceMatchThresholdN;
    BYTE    byNightFaceMatchThresholdN;
    BYTE    byFaceRecognizeEnable;
    BYTE    byBlockFaceMatchThreshold;
    BYTE    byRes3[2];
    BYTE    byDefaultVerifyMode;
    DWORD   dwFingerPrintCapacity;
    DWORD   dwFingerPrintNum;
    BYTE    byEnableFingerPrintNum;
    BYTE    byRes[231];
}NET_DVR_CARD_READER_CFG_V50,*LPNET_DVR_CARD_READER_CFG_V50;
```

## Members

### **dwSize**

Structure size

### **byEnable**

Whether to enable: 0-no, 1-yes.

### **byCardReaderType**

Fingerprint and card reader types: 1-DS-K110XM/MK/C/CK, 7-Wiegand or RS485 offline, 8-DS-K1101M/MK, 9-DS-K1101C/CK, 10-DS-K1102M/MK/M-A, 11-DS-K1102C/CK, 12-DS-K1103M/MK, 13-DS-K1103C/CK, 14-DS-K1104M/MK, 15-DS-K1104C/CK, 16-DS-K1102S/SK/S-A, 19-DS-K1102EM, 20- DS-K1102E, 21-DS-K1200EF, 22-DS-K1200MF, 23-DS-K1200CF, 33-DS-K1T200EF, 34- DS-K1T300EF

### **byOkLedPolarity**

OK LED polarity: 0-negative pole, 1-positive pole.

### **byErrorLedPolarity**

Error LED polarity: 0-negative pole, 1-positive pole.

### **byBuzzerPolarity**

Buzzer polarity: 0-negative pole, 1-positive pole.

### **bySwipeInterval**

Time interval of repeated authentication, which is valid for authentication modes such as fingerprint, card, face, etc., unit: second.

### **byPressTimeout**

Button pressing timeout, unit: second, which is ranging from 1 to 255.

### **byEnableFailAlarm**

Whether to enable excessive failed authentication attempts alarm: 0-no, 1-yes.

### **byMaxReadCardFailNum**

Maximum number of failed authentication attempts, which is ranging from 1 to 10.

### **byEnableTamperCheck**

Whether to enable tampering detection: 0-no, 1-yes.

## **byOfflineCheckTime**

Offline detection time, unit: second, which is ranging from 0 to 255.

## **byFingerPrintCheckLevel**

Fingerprint recognition level: 1-1/10 error rate, 2-1/100error rate, 3-1/1000error rate, 4-1/10000error rate, 5-1/100000error rate, 6-1/1000000error rate, 7-1/10000000error rate, 8-1/100000000error rate, 9-3/100error rate, 10-3/1000error rate, 11-3/10000error rate, 12-3/100000error rate, 13-3/1000000error rate, 14-3/10000000error rate, 15-3/100000000error rate, 16-Auto Normal, 17-Auto Secure, 18-Auto More Secure

## **byUseLocalController**

Read-only, whether is it linked with distributed access controller or not? 0-no, 1-yes.

## **byRes1**

Reserved, set to 0.

## **wLocalControllerID**

Read-only, distributed access controller No. It is valid when **byUseLocalController** is 1, No.0 indicates that the controller is not registered, and the No. is ranging from 1 and 255.

## **wLocalControllerReaderID**

Read-only, fingerprint and card reader No. of distributed access controller. It is valid when **byUseLocalController** is 1, No.0 indicates that the controller is not registered.

## **wCardReaderChannel**

Read-only, communication channel No. of fingerprint an card reader: 0-Wiegand or offline, 1-RS485A, 2-RS485B. It is valid when **byUseLocalController** is 1.

## **byFingerPrintImageQuality**

Fingerprint picture quality: 0-invalid, 1-low (V1), 2-medium (V1), 3-high (V1), 4-highest (V1), 5-low (V2), 6-medium (V2), 7-high (V2), 8-highest (V2).

## **byFingerPrintContrastTimeOut**

Fingerprint picture comparison timeout: 0-invalid, 1 to 20-1 to 20 second, 0xff-unlimited.

## **byFingerPrintRecognizeInterval**

Fingerprint picture comparison interval: 0-invalid, 1 to 10-1 to 10 second, 0xff-no delay.

## **byFingerPrintMatchFastMode**

Fingerprint matching mode: 0-invalid, 1 to 5-fast mode 1 to fast mode 5, 0xff-auto.

## **byFingerPrintModuleSensitive**

Fingerprint module sensitive: 0-invalid, 1 to 8-sensitive level 1 to level 8.

## **byFingerPrintModuleLightCondition**

Fingerprint module light condition: 0-invalid, 1-outdoor, 2-indoor.

## **byFaceMatchThresholdN**

Face picture comparison threshold, which is ranging from 0 to 100.

## **byFaceQuality**

Face picture quality, which is ranging from 0 to 100.

## **byFaceRecognizeTimeOut**

Face recognition timeout: 1 to 20-1s to 20s, 0xff-unlimited.

## **byFaceRecognizeInterval**

Face recognition interval: 0-invalid, 1 to 10-1s to 10s, 0xff-no delay.

## **wCardReaderFunction**

Read-only, fingerprint and card reader types, which is represented by bit: bit1-fingerprint, bit2-face, bit3-pulse; bit value: 0-no, 1-yes

## **byCardReaderDescription**

Fingerprint and card reader description.

## **wFaceImageSensitometry**

Read-only, face picture exposure, which is ranging from 0 to 65535.

## **byLivingBodyDetect**

Live face detection: 0-invalid, 1-disable, 2-disable.

## **byFaceMatchThreshold1**

Face picture 1:1 threshold, which is ranging from 0 to 100.

## **wBuzzerTime**

Buzzing time, which is ranging from 0 to 5999s (0-long buzzing).

## **byFaceMatch1SecurityLevel**

Face picture 1:1 security level: 0-invalid, 1-normal, 2-high, 3-higher

## **byFaceMatchNSecurityLevel**

Face picture 1:N security level: 0-Invalid, 1-normal, 2-high, 3-higher

## **byEnvirMode**

Face recognition environment mode: 0-invalid, 1-indoor, 2-other

## **byLiveDetLevelSet**

Set threshold level of live face detection: 0-invalid, 1-low, 2-medium, 3-high

## **byLiveDetAntiAttackCntLimit**

Anti-attacking times of live face detection: 0-invalid, ranging from 1 to 255.

## **byEnableLiveDetAntiAttack**

Whether to enable the anti-attacking of live face detection: 0-invalid, 1-no, 1-yes.

## **bySupportDelFPByID**

Read-only, whether the fingerprint and card reader supports deleting fingerprint by finger ID: 0-invalid, 1-no, 2-yes.

### **byRes1**

Reserved.

### **byFaceContrastMotionDetLevel**

Motion detection level during face picture comparison: 0-invalid, 1-low, 2-medium, 3-high.

### **byDayFaceMatchThresholdN**

1:N face picture comparison threshold in day, which is between 0 and 100.

### **byNightFaceMatchThresholdN**

1:N face picture comparison threshold at night, which is between 0 and 100.

### **byFaceRecognizeEnable**

Whether to enable facial recognition: 0-invalid, 1-yes (one face), 2-no, 3-yes (multiple faces).

### **byBlockFaceMatchThreshold**

Face picture comparison threshold in blocklist, which is between 0 and 100.

### **byRes3**

Reserved.

### **byDefaultVerifyMode**

Default authentication mode of the fingerprint and card reader (factory settings), read-only: 1-sleepy, 2-card+password, 3-card, 4-card or password, 5-fingerprint, 6-fingerprint+password, 7-fingerprint or card, 8-fingerprint+card, 9-fingerprint+card+password, 10-face or fingerprint or card or password, 11-face+fingerprint, 12-face+password, 13-face+card, 14-face, 15-employee No.+password, 16-fingerprint or password, 17-employee No.+fingerprint, 18-employee No.+fingerprint+password, 19-face+fingerprint+card, 20-face+password+fingerprint, 21-employee No.+face, 22-face or face+card, 23-fingerprint or face, 24-card or face or password, 25-card or face, 26-card or face or fingerprint, 27-card or fingerprint or password.

### **dwFingerPrintCapacity**

Read-only, fingerprint capability, it is valid only when **byEnableFingerPrintNum** is 1.

### **dwFingerPrintNum**

Read-only, number of existing fingerprint pictures, it is valid only when

**byEnableFingerPrintNum** is 1.

### **byEnableFingerPrintNum**

Read-only, whether to enable fingerprint capability: 0-no, 1-yes.

### **byRes**

Reserved, set to 0.

## **A.29 NET\_DVR\_CARD\_READER\_PLAN**

Parameter structure about configuration of authentication mode control schedule.

## Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwTemplateNo;
    BYTE       byRes[64];
}NET_DVR_CARD_READER_PLAN, *LPNET_DVR_CARD_READER_PLAN;
```

## Members

### **dwSize**

Structure size.

### **dwTemplateNo**

Schedule template No.: 0-cancel linking template with schedule, and restore to the default settings (available for swiping card to open the door); non-0-link template with schedule by No.

### **byRes**

Reserved, set to 0.

## A.30 NET\_DVR\_CETTIFICATE\_INFO

Certificate information structure

## Structure Definition

```
struct{
    DWORD          dwSize;
    char           szIssuer[MAX_CERTIFICATE_ISSUER_LEN/*64*/];
    char           szSubject[MAX_CERTIFICATE_SUBJECT_LEN/*64*/];
    NET_DVR_TIME   struStartTime;
    NET_DVR_TIME   struEndTime;
    BYTE          byRes1[1024];
}NET_DVR_CETTIFICATE_INFO, *LPNET_DVR_CETTIFICATE_INFO;
```

## Members

### **dwSize**

Structure size.

### **szIssuer**

Certificate issuer.

### **szSubject**

Certificate holder.

### **struStartTime**

Start time of expiry date, refer to the structure [NET\\_DVR\\_TIME](#) for details.

**struEndTime**

End time of expiry date, refer to the structure [NET\\_DVR\\_TIME](#) for details.

**byRes1**

Reserved.

## A.31 NET\_DVR\_CHECK\_FACE\_PICTURE\_CFG

Structure about face picture verification parameters.

### Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwPictureNo;
    DWORD      dwPictureLen;
    char*      dwPictureLen;;
    DWORD      dwFaceTemplateLen;
    char*      pFaceTemplateBuffer;
    BYTE       byRes[248];
}NET_DVR_CHECK_FACE_PICTURE_CFG, *LPNET_DVR_CHECK_FACE_PICTURE_CFG;
```

### Members

**dwSize**

Structure size.

**dwPictureNo**

Picture No.

**dwPictureLen**

Picture size. The picture should be smaller than or equal to 200 KB.

**pPictureBuffer**

Pointer of the picture.

**dwFaceTemplateLen**

Size of the face modeling data.

**pFaceTemplateBuffer**

Pointer of the face modeling data.

**byRes**

Reserved, set to 0.

## A.32 NET\_DVR\_CHECK\_FACE\_PICTURE\_COND

Structure about condition parameters for verifying face pictures

### Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwPictureNum;
    BYTE       byCheckTemplate;
    BYTE       byRes[127];
}NET_DVR_CHECK_FACE_PICTURE_COND, *LPNET_DVR_CHECK_FACE_PICTURE_COND;
```

### Members

#### **dwSize**

Structure Size.

#### **dwPictureNum**

Number of pictures.

#### **byCheckTemplate**

0-verify whether the face picture is valid (default), 1-verify whether the face picture matches the modeling data.

#### **byRes**

Reserved, set to 0.

## A.33 NET\_DVR\_CHECK\_FACE\_PICTURE\_STATUS

Structure about status and result parameters of face picture verification

### Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwPictureNo;
    BYTE       byCheckStatus;
    BYTE       byRes[127];
}NET_DVR_CHECK_FACE_PICTURE_COND, *LPNET_DVR_CHECK_FACE_PICTURE_COND;
```

### Members

#### **dwSize**

Structure Size.

#### **dwPictureNo**

Picture No.

### **byCheckStatus**

Verification result: 0-invalid, 1-face modeling succeeded, 2-face modeling failed, 3-face module communication exception, 4-there is no face in the picture, 5-face upward, 6-face downward, 7-face left, 8-face right, 9-face rotating clockwise, 10-face rotating anticlockwise, 11-the pupillary distance is too small, 12-the face picture matches the template, 13-the face picture does not match the template, 14-transmission data error.

### **byRes**

Reserved, set to 0.

## **A.34 NET\_DVR\_DATE**

Date information structure.

### **Structure Definition**

```
struct{
    WORD      wYear;
    BYTE      byMonth;
    BYTE      byDay;
}NET_DVR_DATE,*LPNET_DVR_DATE;
```

### **Members**

#### **wYear**

Year

#### **byMonth**

Month

#### **byDay**

Day

## **A.35 NET\_DVR\_DEL\_FINGER\_PRINT\_MODE\_V50**

Parameter union of fingerprint information deleting mode.

### **Structure Union Definition**

```
union{
    BYTE                           uLen[588];
    NET_DVR_FINGER_PRINT_BYCARD_V50 struByCard;
    NET_DVR_FINGER_PRINT_BYREADER_V50 struByReader;
}NET_DVR_DEL_FINGER_PRINT_MODE_V50,*LPNET_DVR_DEL_FINGER_PRINT_MODE_V50;
```

## Members

### **uLen**

Union size.

### **struByCard**

Parameter of deleting fingerprint information by card No. (person ID), see details in the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_BYCARD\\_V50\*\*](#).

### **struByReader**

Parameters of deleting fingerprint information by fingerprint and card reader No., see details in the structure [\*\*NET\\_DVR\\_FINGER\\_PRINT\\_BYREADER\\_V50\*\*](#).

## See Also

[\*\*NET\\_DVR\\_FINGER\\_PRINT\\_INFO\\_CTRL\\_V50\*\*](#)

## A.36 NET\_DVR\_DEVICEINFO\_V30

Device parameter structure (V30).

### Device Parameter Structure (V30)

Member	Data Type	Description
sSerialNumber	BYTE	Device serial No.
byAlarmInPortNum	BYTE	Number of analog alarm inputs
byAlarmOutPortNum	BYTE	Number of analog alarm outputs
byDiskNum	BYTE	Number of HDDs
byDVRTypE	BYTE	Device type
byChanNum	BYTE	Number of analog channels
byStartChan	BYTE	Start No. of analog channel, which starts from 1.
byAudioChanNum	BYTE	Number of two-way audio channels
byIPChanNum	BYTE	Number of digital channels, low 8-bit.
byZeroChanNum	BYTE	Number of channel-zero
byMainProto	BYTE	Transmission protocol type of main stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP

Member	Data Type	Description
bySubProto	BYTE	Transmission protocol type of sub-stream: 0-private protocol (default), 1-RTSP, 2-private protocol+RTSP
bySupport	BYTE	<p>Capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> <li>• bySupport&amp;0x1: whether supports VCA search.</li> <li>• bySupport&amp;0x2: whether supports backup.</li> <li>• bySupport&amp;0x4: whether supports getting encoding parameters.</li> <li>• bySupport&amp;0x8: whether supports dual-NIC.</li> <li>• bySupport&amp;0x10: whether supports remote SADP.</li> <li>• bySupport&amp;0x20: whether supports RAID card.</li> <li>• bySupport&amp;0x40: whether supports searching in IPSAN directory.</li> <li>• bySupport&amp;0x80: whether supports RTP over RTSP.</li> </ul>
bySupport1	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> <li>• bySupport1&amp;0x1: whether supports SNMP with version 30.</li> <li>• bySupport1&amp;0x2: whether supports playback and downloading video files.</li> <li>• bySupport1&amp;0x4: whether supports setting the arming priority.</li> <li>• bySupport1&amp;0x8: whether supports extending the arming time period.</li> <li>• bySupport1&amp;0x10: whether supports multiple HDDs (more than 33).</li> <li>• bySupport1&amp;0x20: whether supports RTP over RTSP.</li> <li>• bySupport1&amp;0x80: whether supports license plate recognition alarm.</li> </ul>

Member	Data Type	Description
bySupport2	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> <li>• bySupport2&amp;0x1: whether supports getting stream via URL.</li> <li>• bySupport2&amp;0x2: whether supports FTP with version 40.</li> <li>• bySupport2&amp;0x4: whether supports ANR.</li> <li>• bySupport2&amp;0x20: whether supports getting device status.</li> <li>• bySupport2&amp;0x40: whether supports encrypting stream.</li> </ul>
wDevType	WORD	Device model
bySupport3	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, while, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> <li>• bySupport3&amp;0x1: whether supports multi-stream.</li> <li>• bySupport3&amp;0x4: whether supports configuring by group (e.g., image, alarm input, alarm output, user, device status, JPEG picture capture, continuous and scheduled capture, .HDD group management, and so on).</li> <li>• bySupport3&amp;0x20: whether supports getting stream via DDNS.</li> </ul>
byMultiStreamProto	BYTE	<p>Whether supports multi-stream, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support.</p> <ul style="list-style-type: none"> <li>• byMultiStreamProto&amp;0x1: whether supports third-stream.</li> <li>• byMultiStreamProto&amp;0x2: whether supports fourth-stream.</li> </ul>

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>• byMultiStreamProto&amp;0x40: whether supports main stream.</li> <li>• byMultiStreamProto&amp;0x80: whether supports sub-stream.</li> </ul>
byStartDChan	BYTE	Start No. of digital channel, 0-no digital channel (e.g., DVR, network camera).
byStartDTalkChan	BYTE	Start No. of two-way audio channel, 0-no two-way audio channel.
byHighDChanNum	BYTE	Number of digital channels, high 8-bit.
bySupport4	BYTE	<p>Extended capabilities, if the result of bitwise operation is 0, it refers that the capability is not supported, if the result is 1, it indicates that the capability is supported.</p> <ul style="list-style-type: none"> <li>• bySupport4&amp;0x01: whether all stream types support RTSP and private protocol.</li> <li>• bySupport4&amp;0x02: whether the device supports transmitting form format data via API (NET_DVR_STDXMLConfig).</li> <li>• bySupport4&amp;0x10: whether supports loading network disk by domain name.</li> </ul>
byLanguageType	BYTE	<p>Supported language types, if the result of bitwise operation is 0, it refers to not support, if the result is 1, it refers to support.</p> <ul style="list-style-type: none"> <li>• byLanguageType ==0: this field is not supported by device.</li> <li>• byLanguageType&amp;0x1: whether supports Chinese.</li> <li>• byLanguageType&amp;0x2: whether supports English.</li> </ul>
byVoiceInChanNum	BYTE	Number of audio input channels
byStartVoiceInChanNo	BYTE	Start No. of audio input channel, 0-invalid.
byRes3	Array of BYTE	Reserved, set to 0.
byMirrorChanNum	BYTE	Number of mirror channels
wStartMirrorChanNo	WORD	Start No. of mirror channel
byRes2	Array of BYTE	Reserved, set to 0.

## Remarks

- The maximum number of digital channels equal to byIPChanNum+byHighDChanNum\*256.
- For login via text protocol, the following parameters are not supported: **byMainProto**, **bySubProto**, **bySupport**, **bySupport1**, **bySupport2**, **bySupport3**, **bySupport4**, **bySupport5**, **bySupport6**, **bySupport7**, **byMultiStreamProto**, **byStartDTalkChan**, **byVoiceInChanNum**, **byStartVoiceInChanNo**, **byMirrorChanNum**, and **wStartMirrorChanNo**.

## See Also

[NET\\_DVR\\_DEVICEINFO\\_V40](#)

## A.37 NET\_DVR\_DEVICEINFO\_V40

### Device Parameter Structure (V40)

Member	Data Type	Description
struDeviceV30	<a href="#"><u>NET_DVR_DEVICEINFO_V30</u></a>	Device parameters
bySupportLock	BYTE	Whether supports locking function: 1-support.
byRetryLoginTime	BYTE	Remaining login attempts, it is valid when the user name or password is incorrect and the <b>bySupportLock</b> is 1.
byPasswordLevel	BYTE	Password strength: 0-invalid, 1-default password, 2-valid password, 3-risky password. For default password or risky password, the users are reminded to change password.
byProxyType	BYTE	Proxy type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
dwSurplusLockTime	DWORD	Remaining locking time, unit: second. It is valid only when <b>bySupportLock</b> is 1. During the locking time, if the user try to log in to again, the remaining locking time will resume to 30 minutes.
byCharEncodeType	BYTE	Character encodings. 0-no decoding information, 1-GB2312 (Simplified Chinese), 2-GBK, 3-BIG5 (Traditional Chinese), 4-Shift_JIS (Japanese), 5-EUC-KR (Korean), 6-UTF-8, 7-

Member	Data Type	Description
		ISO8859-1, 8-ISO8859-2, 9-ISO8859-3, ..., 21-ISO8859-15 (Western European)
bySupportDev5	BYTE	Whether to support getting the parameters of devices that support HCNetSDK version 5.0 or above, the size of device name and type name are extended to 64 bytes.
bySupport	BYTE	Whether it supports uploading changes, it depends on the result of bitwise AND (&) operation: 0-not support, 1-support. The result of <b>bySupport</b> &0x1 indicates that this member is reserved; the result of <b>bySupport</b> &0x2 indicates that whether it supports uploading changes: 0-not support, 1-support. This member is the capability set extension.
byLoginMode	BYTE	Login mode: 0-login via private protocol, 1-login via text protocol. For private protocol, the default login port number is 8000, and for text protocol, the default login port number is 80 or 443.
dwOEMCode	DWORD	OEM code.
iResidualValidity	int	Remaining valid days of the user's password, unit: day. If the negative number is returned, it indicates that the password being used has expired. For example, if -3 is returned, it indicates that the password being used has expired for three days.
byResidualValidity	BYTE	Whether the member <b>iResidualValidity</b> is valid: 0-invalid, 1-valid.
bySingleStartDTalkChan	BYTE	Start channel No. for connecting independent audio tracks to the device. The value 0 is reserved and invalid. The channel No. of audio tracks cannot start from 0.
bySingleDTalkChanNms	BYTE	Total number of channels of the device connected with independent tracks, 0-not support.

Member	Data Type	Description
byPassWordResetLevel	BYTE	Whether to prompt the non-admin user to change the password: 0 (invalid), 1 (If the administrator creates a non-admin user account with an initial password, the non-admin user will be prompted "Please change the initial password" each time he/she logs in to the device until he/she changes the initial password), 2 (If the non-admin user's password has been changed by the administrator, the non-admin user will be prompted "Please set a new password" each time he/she logs in to the device until he/she changes the password).
bySupportStreamEncrypt	BYTE	Whether it supports stream encryption, it depends on the result of bitwise AND (&) operation: 0-no, 1-yes. The result of <b>bySupportStreamEncrypt&amp;0x1</b> indicates whether to support RTP/TLS streaming, the result of <b>bySupportStreamEncrypt&amp;0x2</b> indicates whether to support SRTP/UDP streaming, and the result of <b>bySupportStreamEncrypt&amp;0x4</b> indicates whether to support SRTP/MULTICAST streaming.
byRes2	Array of BYTE	Reserved, set to 0.

## Remarks

- Four character types are allowed in the password, including digits, lowercase letters, uppercase letters and symbols. The maximum password length is 16 bits, and there are four password strength levels, see details below:
  - Level 0 (Risky Password): The password length is less than 8 bits, or only contains one kind of the character types. Or the password is the same with the user name, or is the mirror writing of the user name.
  - Level 1 (Weak Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination should be (digits + lowercase letters) or (digits + uppercase letters).

- Level 2 (Medium Password): The password length is more than or equal to 8 bits, and contains two kinds of the character types. Meanwhile, the combination cannot be (digits + lowercase letters) and (digits + uppercase letters).
- Level 3 (Strong Password): The password length is more than or equal to 8 bits, and at least contains three kinds of the character types.
- For login via text protocol, the following parameters are not supported: **bySupportLock**, **byRetryLoginTime**, **byPasswordLevel**, **byProxyType**, **dwSurplusLockTime**, **byCharEncodeType**, and **bySupportDev5**.

## A.38 NET\_DVR\_DOOR\_CFG

Structure about door (floor) configuration parameters

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byDoorName[DOOR_NAME_LEN/*32*/];
    BYTE       byMagneticType;
    BYTE       byOpenButtonType;
    BYTE       byOpenDuration;
    BYTE       byDisabledOpenDuration;
    BYTE       byMagneticAlarmTimeout;
    BYTE       byEnableDoorLock;
    BYTE       byEnableLeaderCard;
    BYTE       byLeaderCardMode;
    DWORD      dwLeaderCardOpenDuration;
    BYTE       byStressPassword[STRESS_PASSWORD_LEN/*8*/];
    BYTE       bySuperPassword[SUPER_PASSWORD_LEN/*8*/];
    BYTE       byUnlockPassword[UNLOCK_PASSWORD_LEN/*8*/];
    BYTE       byUseLocalController;
    BYTE       byRes1;
    WORD       wLocalControllerID;
    WORD       wLocalControllerDoorNumber;
    WORD       wLocalControllerStatus;
    BYTE       byLockInputCheck;
    BYTE       byLockInputType;
    BYTE       byDoorTerminalMode;
    BYTE       byOpenButton;
    BYTE       byLadderControlDelayTime;
    BYTE       byRes2[43];
}NET_DVR_DOOR_CFG,*LPNET_DVR_CFG;
```

### Members

#### **dwSize**

Structure size.

**byDoorName**

Door (floor) name.

**byMagneticType**

Door magnetic contact type: 0-remain closed, 1-remain open.

**byOpenButtonType**

Exit button type: 0-remain closed, 1-remain open.

**byOpenDuration**

Door open duration (floor relay action duration), value range: [1,255], unit: second.

**byDisabledOpenDuration**

Door open duration for disability cards, value range: [1,255], unit: second.

**byMagneticAlarmTimeout**

Alarm time of door magnetic contact detection timeout, value range: [0,255], unit: second, 0 means no alarm.

**byEnableDoorLock**

Whether to enable locking the door when the door is closed: 0-no, 1-yes.

**byEnableLeaderCard**

Whether to enable remaining the door open with the first card: 0-no, 1-yes.

**byLeaderCardMode**

First card mode: 0-disable, 1-remain open, 2-authorization (**byEnableLeaderCard** is invalid when this member is used).

**dwLeaderCardOpenDuration**

Door open duration with first card, value range: [1,1440], unit: minute.

**byStressPassword**

Duress password.

**bySuperPassword**

Super password.

**byUnlockPassword**

Dismiss code (password for unlocking).

**byUseLocalController**

Whether it is connected to the distributed access controller: 0-no, 1-yes. This member is read-only.

**byRes1**

Reserved, set to 0.

**wLocalControllerID**

Distributed access controller No., it is between 1 and 64, 0 means unregistered. This member is read-only and is valid when **byUseLocalController** is 1.

### wLocalControllerDoorNumber

Door No. of the distributed access controller, it is between 1 and 4 and 0 means unregistered. This member is read-only and is valid when **byUseLocalController** is 1.

### wLocalControllerStatus

Distributed access controller status: 0-offline, 1-online, 2-RS-485 serial port 1 on loop 1, 3-RS-485 serial port 2 on loop 1, 4-RS-485 serial port 1 on loop 2, 5-RS-485 serial port 2 on loop 2, 6-RS-485 serial port 1 on loop 3, 7-RS-485 serial port 2 on loop 3. This member is read-only and is valid when **byUseLocalController** is 1.

### byLockInputCheck

Whether to enable door lock input detection: 0-disable (default), 1-enable.

### byLockInputType

Door lock input type: 0-remain closed (default), 1-remain open.

### byDoorTerminalMode

Door-related terminal working mode: 0-anti-cut and anti-short-circuit (default), 1-normal.

### byOpenButton

Whether to enable the exit button: 0-yes (default), 1-no.

### byLadderControlDelayTime

Visitor delay time of elevator control, value range: [1,255], unit: minute.

### byRes2

Reserved, set to 0.

## A.39 NET\_DVR\_DOOR\_FILE\_UPLOAD\_PARAM

Structure about the parameters of the access control file to be uploaded.

### Structure Definition

```
struct{
    DWORD  dwSize;
    DWORD  dwFileSize;
    BYTE   byFileName[MAX_FILE_NAME_LEN/*100*/];
    BYTE   byRes1[256];
}NET_DVR_DOOR_FILE_UPLOAD_PARAM, *LPNET_DVR_DOOR_FILE_UPLOAD_PARAM;
```

### Members

#### dwSize

Structure size.

**dwFileSize**

File size.

**byFileName**

File name.

**byRes1**

Reserved.

## A.40 NET\_DVR\_DOOR\_STATUS\_PLAN

Parameter structure about door control schedule configuration.

### Structure Definition

```
struct{
    DWORD      dwSize;
    DWORD      dwTemplateNo;
    BYTE       byRes[64];
}NET_DVR_DOOR_STATUS_PLAN, *LPNET_DVR_DOOR_STATUS_PLAN;
```

### Members

**dwSize**

Structure size.

**dwTemplateNo**

Schedule template No.: 0-cancel linking the configured template with schedule, and restore to the default settings; non-0-link the configured template with schedule.

**byRes**

Reserved, set to 0.

## A.41 NET\_DVR\_ETHERNET\_V30

### Ethernet Configuration Structure

Member	Data Type	Description
struDVRIP	<a href="#"><u>NET_DVR_IPADDR_UN ION</u></a>	Device IP address
struDVRIPMask	<a href="#"><u>NET_DVR_IPADDR_UN ION</u></a>	Mask of device IP address

Member	Data Type	Description
dwNetInterface	DWORD	Network interface type: 1-10MBase-T; 2-10MBase-T (full duplex); 3-100MBase-TX; 4-100M (full duplex); 5-10M/100M/1000M (self-adaptive); 6-1000M (full duplex)
wDVRPort	WORD	Device port No.
wMTU	WORD	MTU settings, the default is 1500.
byMACAddr	Array of BYTE	Device physical address.
byEthernetPortNo	BYTE	Network interface No.: 0-invalid, 1-interface 0, 2-interface 1, and so on. This parameter is read-only.
byRes	Array of BYTE	Reserved.

## A.42 NET\_DVR\_EVENT\_CARD\_LINKAGE\_CFG\_V51

Parameter structure about event and card linkage configuration.

### Structure Definition

```
struct{
    DWORD                      dwSize;
    BYTE                       byProMode;
    BYTE                       byRes1[3];
    dwEventSourceID;
    uLinkageInfo;
    byAlarmout[MAX_ALARMHOST_ALARMOUT_NUM/
*512*];
    BYTE                       byRes2[32];
    byOpenDoor[MAX_DOOR_NUM/*256*/];
    byCloseDoor[MAX_DOOR_NUM/*256*/];
    byNormalOpen[MAX_DOOR_NUM/*256*/];
    byNormalClose[MAX_DOOR_NUM/*256*/];
    byMainDevBuzzer;
    byCapturePic;
    byRecordVideo;
    byMainDevStopBuzzer;
    wAudioDisplayID;
    byAudioDisplayMode;
    byRes3[25];
    byReaderBuzzer[MAX_CARD_READER_NUM/*512*/];
    byAlarmOutClose[MAX_ALARMHOST_ALARMOUT_NUM/
*512*];
    BYTE                       byAlarmInSetup[MAX_ALARMHOST_ALARMOUT_NUM/
```

```
*512*/];
    BYTE                                byAlarmInClose [MAX_ALARMHOST_ALARMOUT_NUM/
*512*/];
    BYTE                                byReaderStopBuzzer [MAX_CARD_READER_NUM/
*64*/];
    BYTE                                byRes [512];
}NET_DVR_EVENT_CARD_LINKAGE_CFG_V51, *LPNET_DVR_EVENT_CARD_LINKAGE_CFG_V51;
```

## Members

### **dwSize**

Structure size.

### **byProMode**

Linkage type: 0-event linkage, 1-card No. linkage, 2-MAC address linkage, 3- employee No. (person ID) linkage.

### **byRes1**

Reserved, set to 0.

### **dwEventSourceID**

Event triggering source ID: 0xffffffff-all. For device events, this parameter is invalid; for door events, it refers to door No.; for card reader events, it refers to card reader ID; for alarm input events, it refers to zone or event alarm input ID.

### **uLinkageInfo**

Linkage action parameter, see [\*\*NET\\_DVR\\_EVENT\\_CARD\\_LINKAGE\\_UNION\*\*](#) for details.

### **byAlarmout**

Linked alarm output No., which is represented by byte. 0-not link, 1-link.

### **byRes2**

Reserved, set to 0.

### **byOpenDoor**

Whether to enable door opening linkage, which is represented by byte. 0-disable, 1-enable.

### **byCloseDoor**

Whether to enable door closing linkage, which is represented by byte. 0-disable, 1-enable.

### **byNormalOpen**

Whether to enable door remaining open linkage, which is represented by byte. 0-disable, 1-enable.

### **byNormalClose**

Whether to enable door remaining closed linkage, which is represented by byte. 0-disable, 1-enable.

### **byMainDevBuzzer**

Whether to enable access controller buzzing, 0-disable, 1-enable.

### **byCapturePic**

Whether to enable capture linkage, 0-disable, 1-enable.

### **byRecordVideo**

Whether to enable recording linkage, 0-disable, 1-enable.

### **byMainDevStopBuzzer**

Whether to enable access controller stopping buzzing linkage, 0-disable, 1-enable.

### **wAudioDisplayID**

Linked audio prompt ID, currently it is between 1 and 32, and 0 indicates no linkage.

### **byAudioDisplayStyle**

Linked audio prompt mode: 0-disable, 1-play once, 2-loop playing.

### **byRes3**

Reserved.

### **byReaderBuzzer**

Whether to enable buzzer linkage, which is represented by byte. 0-disable, 1-enable.

### **byAlarmOutClose**

Whether to enable alarm output disabling linkage, which is represented by byte. 0-disable, 1-enable.

### **byAlarmInSetup**

Whether to enable zone arming linkage, which is represented by byte. 0-disable, 1-enable.

### **byAlarmInClose**

Whether to enable zone disarming linkage, which is represented by byte. 0-disable, 1-enable.

### **byReaderStopBuzzer**

Whether to enable card reader stopping buzzing linkage, which is represented by byte. 0-disable, 1-enable.

### **byRes**

Reserved, set to 0.

## **A.43 NET\_DVR\_EVENT\_CARD\_LINKAGE\_COND**

Condition structure about the event card linkage configuration.

### **Structure Definition**

```
struct{
    DWORD      dwSize;
    DWORD      dwEventID;
    WORD       wLocalControllerID;
```

```
BYTE    byRes[106];
}NET_DVR_EVENT_CARD_LINKAGE_COND, *LPNET_DVR_EVENT_CARD_LINKAGE_COND;
```

### Members

#### **dwSize**

Structure size.

#### **dwEventID**

Event ID.

#### **wLocalControllerID**

Distributed access controller No. which is between 1 and 64, while, 0-access controller.

#### **byRes**

Reserved, set to 0.

## A.44 NET\_DVR\_EVENT\_LINKAGE\_INFO

Event linkage parameter structure.

### Structure Definition

```
struct{
    WORD    wMainEventType;
    WORD    wSubEventType;
    BYTE    byRes[28];
}NET_DVR_EVENT_LINKAGE_INFO, *LPNET_DVR_EVENT_LINKAGE_INFO;
```

### Members

#### **wMainEventType**

Event major types, see [\*\*Access Control Event Types\*\*](#) for details.

#### **wSubEventType**

Event minor types, see [\*\*Access Control Event Types\*\*](#) for details.

#### **byRes**

Reserved, set to 0.

### See Also

[NET\\_DVR\\_EVETN\\_CARD\\_LINKAGE\\_UNION](#)

## A.45 NET\_DVR\_EVETN\_CARD\_LINKAGE\_UNION

Parameter union about event and card linkage configuration.

## Structure Definition

```
union{
    BYTE           byCardNo [ACS_CARD_NO_LEN/*32*/];
    NET_DVR_EVENT_LINKAGE_INFO struEventLinkage;
    BYTE           byMACAddr [MACADDR_LEN/*6*/];
    BYTE           byEmployeeNo [NET_SDK_EMPLOYEE_NO_LEN/*32*/];
}NET_DVR_EVETN_CARD_LINKAGE_UNION,*LPNET_DVR_EVETN_CARD_LINKAGE_UNION;
```

## Members

### **byCardNo**

Card No.

### **struEventLinkage**

Event linkage parameters, see details in the structure [NET\\_DVR\\_EVENT\\_LINKAGE\\_INFO](#).

### **byMACAddr**

Physical MAC address.

### **byEmployeeNo**

Employee No. (person ID)

## See Also

[NET\\_DVR\\_EVENT\\_CARD\\_LINKAGE\\_CFG\\_V51](#)

## A.46 NET\_DVR\_FACE\_FEATURE

Structure about facial feature parameters.

## Structure Definition

```
struct{
    NET_VCA_RECT   struFace;
    NET_VCA_POINT  struLeftEye;
    NET_VCA_POINT  struRightEye;
    NET_VCA_POINT  struLeftMouth;
    NET_VCA_POINT  struRightMouth;
    NET_VCA_POINT  struNoseTip;
}NET_DVR_FACE_FEATURE, *LPNET_DVR_FACE_FEATURE;
```

## Members

### **struFace**

Face sub-picture area, see details in the structure [NET\\_VCA\\_RECT](#).

### **struLeftEye**

Coordinates of the left eye, see details in the structure [NET\\_VCA\\_POINT](#).

### **struRightEye**

Coordinates of the right eye, see details in the structure [NET\\_VCA\\_POINT](#).

### **struLeftMouth**

Coordinates of the left mouth corner, see details in the structure [NET\\_VCA\\_POINT](#).

### **struRightMouth**

Coordinates of the right mouth corner, see details in the structure [NET\\_VCA\\_POINT](#).

### **struNoseTip**

Coordinates of the nose, see details in the structure [NET\\_VCA\\_POINT](#).

## A.47 NET\_DVR\_FAILED\_FACE\_COND

Condition structure of getting face picture information, of which modeling failed

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byRes [128];
}NET_DVR_FAILED_FACE_COND, *LPNET_DVR_FAILED_FACE_COND;
```

### Members

#### **dwSize**

Structure size.

#### **byRes**

Reserved, set to 0

## A.48 NET\_DVR\_FAILED\_FACE\_INFO

Structure of modeling failure information parameters

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byCardNo [ACS_CARD_NO_LEN/*32*/];
    BYTE       byErrorCode;
    BYTE       byRes1 [3];
    BYTE       byEmployeeNo [NET_SDK_EMPLOYEE_NO_LEN/*32*/];
    BYTE       byRes [92];
}NET_DVR_FAILED_FACE_INFO, *LPNET_DVR_FAILED_FACE_INFO;
```

### Members

#### **dwSize**

Structure size.

#### **byCardNo**

No. of the card linked to the face.

#### **byErrorCode**

Failure error code: 0-Invalid, 1-reading picture file failed, 2-opening picture file failed, 3-no enough memory, 4-face modeling failed, 5-the pupillary distance is too small (less than 60), 6-the card has no permission.

#### **byRes1**

Reserved.

#### **byEmployeeNo**

Employee No. (person ID)

#### **byRes**

Reserved, set to 0.

## A.49 NET\_DVR\_FINGER\_PRINT\_BYCARD\_V50

Parameter structure of deleting fingerprint information by card No. (person ID).

### Structure Definition

```
struct{
    BYTE  byCardNo [ACS_CARD_NO_LEN/*32*/];
    BYTE  byEnableCardReader [MAX_CARD_READER_NUM/*512*/];
    BYTE  byFingerPrintID [MAX_FINGER_PRINT_NUM/*10*/];
    BYTE  byRes1[2];
    BYTE  byEmployeeNo [NET_SDK_EMPLOYEE_NO_LEN/*32*/];
}NET_DVR_FINGER_PRINT_BYCARD_V50,*LPNET_DVR_FINGER_PRINT_BYCARD_V50;
```

### Members

#### **byCardNo**

No. of card linked with the fingerprint information.

#### **byEnableCardReader**

Fingerprint and card reader IDs, which are represented by array, and each member of array refers to one reader ID, value: 0-not delete, 1-delete.

#### **byFingerPrintID**

Finger IDs, which are represented by array, and each member of the array refers to one fingerprint ID, value: 0-not delete, 1-delete.

### **byRes1**

Reserved.

### **byEmployeeNo**

Employee ID (person ID).

## See Also

### [NET\\_DVR\\_DEL\\_FINGER\\_PRINT\\_MODE\\_V50](#)

## A.50 NET\_DVR\_FINGER\_PRINT\_BYREADER\_V50

Parameter structure of deleting fingerprint information by fingerprint and card reader No.

### Structure Definition

```
struct{
    DWORD dwCardReaderNo;
    BYTE byClearAllCard;
    BYTE byRes1[3];
    BYTE byCardNo[ACS_CARD_NO_LEN/*32*/];
    BYTE byEmployeeNo[NET_SDK_EMPLOYEE_NO_LEN/*32*/];
    BYTE byRes[516];
}NET_DVR_FINGER_PRINT_BYREADER_V50,*LPNET_DVR_FINGER_PRINT_BYREADER_V50;
```

## Members

### **dwCardReaderNo**

Fingerprint and card reader No.

### **byClearAllCard**

Whether to delete the fingerprint information of all cards: 0-no, delete by card No. (person ID);  
1-yes.

### **byRes1**

Reserved.

### **byCardNo**

No. of card linked with the fingerprint information

### **byEmployeeNo**

Employee ID (person ID)

### **byRes**

Reserved.

## See Also

[NET\\_DVR\\_DEL\\_FINGER\\_PRINT\\_MODE\\_V50](#)

## A.51 NET\_DVR\_FINGER\_PRINT\_CFG\_V50

Fingerprint configuration structure.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byCardNo[ACS_CARD_NO_LEN/*32*/];
    DWORD      dwFingerPrintLen;
    BYTE       byEnableCardReader[MAX_CARD_READER_NUM/*512*/];
    BYTE       byFingerPrintID;
    BYTE       byFingerType;
    BYTE       byRes1[30];
    BYTE       byFingerData[MAX_FINGER_PRINT_LEN/*768*/];
    BYTE       byEmployeeNo[NET_SDK_EMPLOYEE_NO_LEN/*32*/];
    BYTE       byLeaderFP[MAX_DOOR_NUM_256/*256*/];
    BYTE       byRes[128];
}NET_DVR_FINGER_PRINT_CFG, *LPNET_DVR_FINGER_PRINT_CFG;
```

### Members

#### dwSize

Structure size.

#### byCardNo

Card No., which is linked with the fingerprint.

#### dwFingerPrintLen

Size of fingerprint data. The fingerprint module and fingerprint recorder will be used in pair.

#### byEnableCardReader

Whether to apply fingerprint data to the fingerprint module, which is represented by array: 0-no  
1-yes

#### byFingerPrintID

Finger No., which is between 1 and 10

#### byFingerType

Fingerprint type: 0-normal fingerprint, 1-duress fingerprint, 2-patrol fingerprint, 3-super  
fingerprint, 4-dismiss fingerprint

#### byRes1

Reserved, set to 0

## **byFingerData**

Fingerprint data

## **byEmployeeNo**

Employee No. (person ID)

## **byLeaderFP**

Whether to support first time authentication function (door, represented by byte): 0-no, 1-yes.

## **byRes**

Reserved, set to 0

## **A.52 NET\_DVR\_FINGER\_PRINT\_INFO\_COND\_V50**

Fingerprint parameter configuration structure.

### **Structure Definition**

```
struct{
    DWORD      dwSize;
    BYTE       byCardNo[ACS_CARD_NO_LEN/*32*/];
    BYTE       byEnableCardReader[MAX_CARD_READER_NUM/*512*/];
    DWORD      dwFingerPrintNum;
    BYTE       byFingerPrintID;
    BYTE       byCallbackMode;
    BYTE       byRes2[2];
    BYTE       byEmployeeNo[NET_SDK_EMPLOYEE_NO_LEN/*32*/];
    BYTE       byRes1[128];
}NET_DVR_FINGER_PRINT_INFO_COND_V50, *LPNET_DVR_FINGER_PRINT_INFO_COND_V50;
```

### **Members**

#### **dwSize**

Structure size.

#### **byCardNo**

Card No. linked with the fingerprint. This parameter is invalid when setting fingerprint parameters.

#### **byEnableCardReader**

Fingerprint module status: 0-invalid; 1-valid.

#### **dwFingerPrintNum**

Number of obtained or configured fingerprints, 0xffffffff-get all fingerprints' information.

#### **byFingerPrintID**

Finger No., which is between 1 and 10, 0xff indicates-all fingerprints of the card.

#### **byCallbackMode**

Device callback mode: 0Returned when applied all; 1Returned when applied a part.

### **byRes2**

Reserved, set to 0.

### **byEmployeeNo**

Employee No. (person ID).

### **byRes1**

Reserved, set to 0.

## Remarks

Two fingerprint applying modes are available: blocking mode and non-blocking mode.

- Blocking Mode: Set **byCallbackMode** to "0", and the applying status will be returned for once only after applying each fingerprint.
- Non-blocking Mode: Set **byCallbackMode** to "1", and the applying status will be returned for multiple times after applying each fingerprint. And the next fingerprint can be applied until the previous fingerprint information is applied.

## A.53 NET\_DVR\_FINGER\_PRINT\_INFO\_CTRL\_V50

Condition structure for deleting fingerprint information.

### Structure Definition

```
struct{
    WORD dwSize;
    BYTE byMode;
    BYTE byRes1[3];
    NET_DVR_DEL_FINGER_PRINT_MODE_V50 struProcessMode;
    BYTE byRes[64];
}NET_DVR_FINGER_PRINT_INFO_CTRL_V50,*LPNET_DVR_FINGER_PRINT_INFO_CTRL_V50;
```

## Members

### **dwSize**

Structure size.

### **byMode**

Deleting mode: 0-by card No. (person ID), 1-by fingerprint and card reader No.

### **byRes1**

Reserved.

### **struProcessMode**

Deleting mode parameters, refer to the data union [NET\\_DVR\\_DEL\\_FINGER\\_PRINT\\_MODE\\_V50](#) for details.

**byRes**

Reserved.

## A.54 NET\_DVR\_FINGER\_PRINT\_STATUS\_V50

Fingerprint information applying or deleting status structure.

### Structure Definition

```
struct{
    WORD dwSize;
    BYTE byCardNo[ACS_CARD_NO_LEN/*32*/];
    BYTE byCardReaderRecvStatus[MAX_CARD_READER_NUM_512/*512*/];
    BYTE byFingerPrintID;
    BYTE byFingerType;
    BYTE byTotalStatus;
    BYTE byRecvStatus;
    BYTE byErrorMsg[ERROR_MSG_LEN/*32*/];
    WORD dwCardReaderNo;
    BYTE byEmployeeNo[NET_SDK_EMPLOYEE_NO_LEN/*32*/];
    BYTE byErrorEmployeeNo[NET_SDK_EMPLOYEE_NO_LEN/*32*/];
    BYTE byRes[128];
}NET_DVR_FINGER_PRINT_STATUS_V50,*LPNET_DVR_FINGER_PRINT_STATUS_V50;
```

### Members

**dwSize**

Structure size.

**byCardNo**

Card No. which is linked with the fingerprint.

**byCardReaderRecvStatus**

Fingerprint module status, which is represented by byte: 0-failed, 1-completed, 2-the fingerprint module is offline, 3-try again or poor fingerprint quality, 4-memory is full, 5-the fingerprint already exists, 6-the fingerprint ID already exists, 7-invalid fingerprint ID, 8-the fingerprint module does not need to be configured, 10-the fingerprint module version is too old to support the employee No.

**byFingerPrintID**

Finger No. which is between 1 and 10.

**byFingerType**

Fingerprint type: 0-normal fingerprint, 1-duress fingerprint, 2-patrol fingerprint, 3-super fingerprint, 4-dismiss fingerprint.

**byTotalStatus**

Applying status: 0-applying, 1-applied to all (it does not mean that all the applying succeeded).

## **byRecvStatus**

Device error status: 0-succeeded, 1-invalid finger No., 2-invalid fingerprint type, 3-invalid card No. (the card No. does not meet the device requirements), 4-the fingerprint is not linked to employee No. or card No. (the employee No. or the card No. is NULL), 5-the employee No. does not exist, 6-the fingerprint data length is 0, 7-invalid card reader No., 8-invalid employee No., 9-illegal first time authentication value, 10-other parameters error.

## **byErrorMsg**

Apply error message. When **byCardReaderRecvStatus** is 5, it indicates that the card No. linked to the fingerprint already exists.

## **dwCardReaderNo**

When **byCardReaderRecvStatus** is 5, it indicates that the fingerprint module corresponding to the fingerprint already exists. This member can be used to return applying error, and 0 refers to no error message.

## **byEmployeeNo**

Employee No. (person ID)

## **byErrorEmployeeNo**

Apply error message. When **byCardReaderRecvStatus** is 5, it indicates that the employee No. (person ID) linked to the fingerprint already exists.

## **byRes**

Reserved.

## **Related API**

### [NET\\_DVR\\_StartRemoteConfig](#)

## **A.55 NET\_DVR\_GROUP\_CFG**

Group configuration structure.

### **Structure Definition**

```
struct{
    DWORD             dwSize;
    BYTE              byEnable;
    BYTE              byRes1[3];

    NET_DVR_VALID_PERIOD_Cfg
    struValidPeriodCfg;
    BYTE              byGroupName[GROUP_NAME_LEN/*32*/];
    BYTE              byRes2[32];
}NET_DVR_GROUP_CFG, *LPNET_DVR_GROUP_CFG;
```

### Members

#### **dwSize**

Structure size

#### **byEnable**

Whether to enable the group: 0-no, 1-yes.

#### **byRes1**

Reserved, set to 0.

#### **struValidPeriodCfg**

Group expiry date.

#### **byGroupName**

Group name

#### **byRes2**

Reserved, set to 0.

## A.56 NET\_DVR\_GROUP\_COMBINATION\_INFO\_V50

Group parameters structure.

### Structure Definition

```
struct{
    BYTE      byEnable;
    BYTE      byMemberNum;
    BYTE      bySequenceNo;
    BYTE      byRes;
    DWORD    dwGroupNo;
}NET_DVR_MULTI_CARD_CFG_V50,*LPNET_DVR_MULTI_CARD_CFG_V50;
```

### Members

#### **byEnable**

Whether to enable the group: 0-no, 1-yes

#### **byMemberNum**

Number of cards should be swiped in the group.

#### **bySequenceNo**

Card swiping order in the group.

#### **byRes**

Reserved, set to 0.

#### **dwGroupNo**

Group No., 0xffffffff-remotely open door, 0xffffffe-open door by super password.

### See Also

[NET\\_DVR\\_MULTI\\_CARD\\_GROUP\\_CFG\\_V50](#)

## A.57 NET\_DVR\_HOLIDAY\_GROUP\_CFG

Holiday group configuration structure.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byEnable;
    BYTE       byRes1[3];
    BYTE       byGroupName[HOLIDAY_GROUP_NAME_LEN/*32*/];
    DWORD      dwHolidayPlanNo[MAX_HOLIDAY_PLAN_NUM/*16*/];
    BYTE       byRes2[32];
}NET_DVR_HOLIDAY_GROUP_CFG,*LPNET_DVR_HOLIDAY_GROUP_CFG;
```

### Members

#### **dwSize**

Structure size.

#### **byEnable**

Whether to enable: 1-enable, 0-disable.

#### **byRes1**

Reserved, set to 0.

#### **byGroupName**

Holiday group name.

#### **dwHolidayPlanNo**

Holiday schedule No.: 0-invalid.

#### **byRes2**

Reserved, set to 0.

## A.58 NET\_DVR\_HOLIDAY\_PLAN\_CFG

Holiday schedule configuration structure.

## Structure Definition

```

struct{
    DWORD dwSize;
    BYTE byEnable;
    BYTE byRes1[3];

    NET_DVR_DATE struBeginDate;

    NET_DVR_DATE struEndDate;

    NET_DVR_SINGLE_PLAN_SEGMENT
        struPlanCfg[MAX_DAYS][MAX_TIMESEGMENT_V30];
    BYTE byRes2[16];
}NET_DVR_HOLIDAY_PLAN_Cfg,*LPNET_DVR_HOLIDAY_PLAN_Cfg;

```

## Members

### **dwSize**

Structure size.

### **byEnable**

Enable? 0- No; 1- Yes

### **byRes1**

Reserved, set to 0.

### **struBeginDate**

Holiday start time, see **NET\_DVR\_DATE** for details.

### **struEndDate**

Holiday end time, see **NET\_DVR\_DATE** for details.

### **struPlanCfg**

Holiday schedule parameters, up to 7 days can be set in one week, and up to 8 time periods can be set in one day, see **NET\_DVR\_SINGLE\_PLAN\_SEGMENT** for details.

### **byRes2**

Reserved, set to 0.

## A.59 NET\_DVR\_ID\_CARD\_INFO

## Structure about ID Card Information

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>byName</b>	Array [BYTE]	Name. The maximum size is 128 bytes (the value of the macro definition "MAX_ID_NAME_LEN").
<b>struBirth</b>	<b><u>NET_DVR_DATE</u></b>	Date of birth.
<b>byAddr</b>	Array [BYTE]	Address. The maximum size is 280 bytes (the value of the macro definition "MAX_ID_ADDR_LEN").
<b>byIDNum</b>	Array [BYTE]	ID Card No. The maximum size is 32 bytes (the value of the macro definition "MAX_ID_NUM_LEN").
<b>byIssuingAuthority</b>	Array [BYTE]	Issuing authority. The maximum size is 128 bytes (the value of the macro definition "MAX_ID_ISSUING_AUTHORITY_LEN").
<b>struStartDate</b>	<b><u>NET_DVR_DATE</u></b>	Start data of the effective period.
<b>struEndDate</b>	<b><u>NET_DVR_DATE</u></b>	End data of the effective period.
<b>byTermOfValidity</b>	BYTE	Whether it is permanently valid: 0-no, 1-yes (the end date of the effective period is invalid).
<b>bySex</b>	BYTE	Gender: 1-male, 2-female.
<b>byRes0</b>	BYTE	Reserved.
<b>byRes</b>	Array [BYTE]	Reserved. The maximum size is 101 bytes.

## A.60 NET\_DVR\_ID\_CARD\_INFO\_ALARM

### Structure about Alarm Triggered by Swiping ID Card

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>struIDCardCfg</b>	<b><i>NET_DVR_ID_CARD_INFO</i></b>	ID card information.
<b>dwMajor</b>	DWORD	Major alarm types, see details in <a href="#"><b>Access Control Event Types</b></a> .
<b>dwMinor</b>	DWORD	Minor alarm types, see details in <a href="#"><b>Access Control Event Types</b></a> .
<b>struSwipeTime</b>	<b><i>NET_DVR_TIME_V30</i></b>	Card swiping time.
<b>byNetUser</b>	Array [BYTE]	User name for network operations. The maximum size is 16 bytes (the value of the macro definition "MAX_NAMELEN").
<b>struRemoteHostAddr</b>	<b><i>NET_DVR_IPADDR_UNION</i></b>	IP address of the remote access controller.
<b>dwCardReaderNo</b>	DWORD	Card reader No., 0-invalid.
<b>dwDoorNo</b>	DWORD	Door No., 0-invalid. For the turnstile (swing barrier), 1 refers to entrance and 2 refers to exit.
<b>dwPicDataLen</b>	DWORD	Picture data size. If this member is not 0, it indicates the picture data exists.
<b>pPicData</b>	char*	Pointer of the picture data.
<b>byCardType</b>	BYTE	Card type: 1-normal card, 2-disability card, 3-card in the blocklist, 4-patrol card, 5-duress card, 6-super card, 7-visitor card, 8-dismiss card, 0-invalid

Member	Data Type	Description
<b>byDeviceNo</b>	BYTE	Device No., it is between 1 and 255, 0-invalid.
<b>byMask</b>	BYTE	Whether the person is wearing mask: 0-reserved, 1-unknown, 2-not wearing mask, 3-wearing mask.
<b>byCurrentEvent</b>	BYTE	Whether it is a real-time event: 0-invalid, 1-yes (real-time event), 2-no (offline event).
<b>dwFingerPrintDataLen</b>	DWORD	Fingerprint data size. If this member is not 0, it indicates that the fingerprint data exists.
<b>pFingerPrintData</b>	char*	Fingerprint data.
<b>dwCapturePicDataLen</b>	DWORD	Data size of the captured picture. If this member is not 0, it indicates that the captured picture data exists.
<b>pCapturePicData</b>	char*	Captured picture data.
<b>dwCertificatePicDataLen</b>	DWORD	Data size of the captured certificate picture. If this member is not 0, it indicates that the captured certificate picture data exists.
<b>pCertificatePicData</b>	char*	Captured certificate picture data.
<b>byCardReaderKind</b>	BYTE	Reader type: 0-invalid, 1-IC card reader, 2-ID card reader, 3-QR code scanner, 4-fingerprint module.
<b>byHelmet</b>	BYTE	Whether the person is wearing a hard hat: 0-reserved, 1-unknown, 2-not wearing a hard hat, 3-wearing a hard hat.
<b>byRes3</b>	BYTE	Reserved.

Member	Data Type	Description
<b>byIDCardInfoExtend</b>	BYTE	Whether the member <b>pIDCardInfoExtend</b> is valid: 0-invalid, 1-valid.
<b>pIDCardInfoExtend</b>	char*	When <b>byIDCardInfoExtend</b> is set to 1, it points to the structure <b>NET_DVR_ID_CARD_INFO_EXTEND</b> .
<b>dwSerialNo</b>	DWORD	Event serial No., 0 means invalid.
<b>byHealthCode</b>	BYTE	Health code status: 0 (no request), 1 (no health code), 2 (green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6 (other error, e.g., searching failed due to API exception), 7 (searching for the health code timed out).
<b>byRes</b>	Array [BYTE]	Reserved. The size is 167 bytes.

## A.61 NET\_DVR\_ID\_CARD\_INFO\_EXTEND

### Structure about Extended ID Card Information

Member	Data Type	Description
<b>byRemoteCheck</b>	BYTE	Whether remote verification is required: 0-invalid, 1-no (default), 2-yes.
<b>byThermometryUnit</b>	BYTE	Temperature unit: 0-Celsius (default), 1-Fahrenheit, 3-Kelvin.
<b>byIsAbnormalTemperature</b>	BYTE	Whether the face temperature is abnormal: 1-yes, 0-no.
<b>byRes2</b>	BYTE	Reserved.

Member	Data Type	Description
<b>fCurrTemperature</b>	float	Face temperature, it is accurate to one decimal place.
<b>struRegionCoordinates</b>	<b><i>NET_VCA_POINT</i></b>	Face temperature's coordinates.
<b>dwQRCodeInfoLen</b>	DWORD	Data size of the QR code information. If this member is not 0, it indicates that the QR code information data exists.
<b>dwVisibleLightDataLen</b>	DWORD	Data size of the visible light picture captured by the thermal camera. If this member is not 0, it indicates that the visible light picture data exists.
<b>dwThermalDataLen</b>	DWORD	Data size of the thermal picture. If this member is not 0, it indicates that the thermal picture data exists.
<b>pQRCodeInfo</b>	char*	Pointer of the QR code information.
<b>pVisibleLightData</b>	char*	Pointer of the visible light picture captured by the thermal camera.
<b>pThermalData</b>	char*	Pointer of the thermal picture.
<b>byRes</b>	Array [BYTE]	Reserved. The maximum size is 1024 bytes.

## A.62 NET\_DVR\_INIT\_CFG\_ABILITY

### Initialization Capability Structure

Member	Data Type	Description
enumMaxLoginUsersNum	INIT_CFG_MAX_NUM	Maximum number of users can log in, see details below:

Member	Data Type	Description
		<pre>enum _INIT_CFG_MAX_NUM_ {     INIT_CFG_NUM_2048 = 2048,     INIT_CFG_NUM_5120 = 5120,     INIT_CFG_NUM_10240 = 10240,     INIT_CFG_NUM_15360 = 15360,     INIT_CFG_NUM_20480 = 20480 } INIT_CFG_MAX_NUM</pre>
enumMaxAlarmNum	INIT_CFG_MAX_NUM	<p>Maximum number of alarm channels, see details below:</p> <pre>enum _INIT_CFG_MAX_NUM_ {     INIT_CFG_NUM_2048 = 2048,     INIT_CFG_NUM_5120 = 5120,     INIT_CFG_NUM_10240 = 10240,     INIT_CFG_NUM_15360 = 15360,     INIT_CFG_NUM_20480 = 20480 } INIT_CFG_MAX_NUM</pre>
byRes	Array of BYTE	Reserved, set to 0.

## Remarks

By default, up to 2048 channels are supported. More channels require higher computer performance and network bandwidth.

## See Also

[NET\\_DVR\\_SetSDKInitCfg](#)

## A.63 NET\_DVR\_IPADDR\_UNION

### IP Address Union

Member	Data Type	Description
sIPv4	char[]	IPv4 address. The maximum length is 16 bytes.
sIPv6	char[]	IPv6 address. The maximum length is 256 bytes.

## A.64 NET\_DVR\_JSON\_DATA\_CFG

Structure about picture data in JSON format.

## Structure Definition

```
struct{
    DWORD      dwSize;
    void       *lpJsonData;
    DWORD      dwJsonDataSize;
    void       *lpPicData;
    DWORD      dwPicDataSize;
    DWORD      dwInfraredFacePicSize;
    char       *lpInfraredFacePicBuffer;
    BYTE       byRes[248];
}NET_DVR_JSON_DATA_CFG,*LPNET_DVR_JSON_DATA_CFG;
```

## Members

### **dwSize**

Structure size.

### **lpjsonData**

Returned message in JSON format.

### **dwJsonDataSize**

Size of the message in JSON format.

### **lpPicData**

Picture data. If the returned message is the response status message, this member is invalid; if the returned message in JSON format does not contain **faceURL**, this member should contain picture data in binary format.

### **dwPicDataSize**

Picture data size, the maximum size is 200 KB.

### **dwInfraredFacePicSize**

Data size of the infrared face picture. When this member is 0, it indicates that there is no face picture data. When the response message is **JSONResponseStatus**, this member is meaningless. When the request message in JSON format does not contain the value of **infraredFaceURL**, this member should contain the binary picture.

### **lpInfraredFacePicBuffer**

Buffer of infrared face picture data.

### **byRes**

Reserved.

## A.65 NET\_DVR\_LOCAL\_ABILITY\_PARSE\_CFG

Structure about capability of analysis library configuration.

## Structure Definition

```
struct{
    BYTE      byEnableAbilityParse;
    BYTE      byRes[127];
}NET_DVR_LOCAL_ABILITY_PARSE_CFG, *LPNET_DVR_LOCAL_ABILITY_PARSE_CFG;
```

## Members

### **byEnableAbilityParse**

Whether to enable capability analysis library: 0-disable, 1-enable (default).

### **byRes**

Reserved, set to 0.

## Remarks

By default, the analog capability is disabled, you can enable the analog capability via this structure, and then call [\*\*NET\\_DVR\\_GetDeviceAbility\*\*](#) and load the "LocalXml.zip" to the directory of HCNetSDK to get the capabilities of devices.

## A.66 NET\_DVR\_LOCAL\_ASYNC\_CFG

### Structure about Asynchronous Configuration Parameter

Member	Data Type	Description
<b>bEnable</b>	BOOL	Whether to enable asynchronous configuration: "TRUE"-yes, "FALSE"-no (default).
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 60 bytes.

## Remarks

- After enabling asynchronous configuration, the notifications about disconnection and reconnection of devices will be received in asynchronous mode. This function can be adopted when you need to manage tens of thousands of devices. By default, this function is disabled.
- After enabling asynchronous configuration, the interval configuration of heartbeat interaction turns invalid (related command: "NET\_SDK\_LOCAL\_CFG\_TYPE\_CHECK\_DEV").
- After enabling asynchronous configuration, the API [\*\*NET\\_DVR\\_SetConnectTime\*\*](#) for setting network connection timeout turns invalid.

## A.67 NET\_DVR\_LOCAL\_BYTE\_ENCODE\_CONVERT

Structure about encoding format conversion configuration.

### Structure Definition

```
struct{
    CHAR_ENCODE_CONVERT      fnCharConvertCallBack
    BYTE                      byRes [256];
}NET_DVR_LOCAL_BYTE_ENCODE_CONVERT, *LPNET_DVR_LOCAL_BYTE_ENCODE_CONVERT;
```

### Members

#### fnCharConvertCallBack

Callback function of encoding type conversion, see details in [CHAR\\_ENCODE\\_CONVERT](#).

#### byRes

Reserved, set to 0.

### Remarks

- The device character encoding type is returned by the login API.
- By default, the encoding type conversion is realized by the "libiconv.dll" of HCNetSDK, but the users can set the encoding type conversion callback via this structure and call their own encoding API to convert the encoding type.

## A.68 NET\_DVR\_LOCAL\_CERTIFICATION

Certificate configuration parameter structure

### Structure Definition

```
struct{
    char                      szLoadPath[MAX_FILE_PATH_LEN/*256*/];
    fnCertVerifyResultCallBack fnCB;
    void                      *pUserData;
    BYTE                      byRes [64];
}NET_DVR_LOCAL_CERTIFICATION, *LPNET_DVR_LOCAL_CERTIFICATION;
```

### Members

#### szLoadPath

Certificate saving path.

#### fnCB

Certificate verification callback function, see details below.

```
typedef BOOL (CALLBACK *fnCertVerifyResultCallBack) (
    DWORD uiResult,
    NET_DVR_CETTIFICATE_INFO lpCertificateInfo,
    char *pUserData
);
```

### **uiResult**

Certificate verification results: 0-verification failed, other values-verified.

### **lpCertificateInfo**

Certificate information, see details in [NET\\_DVR\\_CETTIFICATE\\_INFO](#).

### **pUserData**

User data pointer.

### **pUserData**

User data.

### **byRes**

Reserved, set to 0.

## See Also

[NET\\_SDK\\_LOCAL\\_CFG\\_TYPE](#)

## A.69 NET\_DVR\_LOCAL\_CFG\_TYPE\_PTZ

PTZ interaction configuration structure.

## Structure Definition

```
struct{
    BYTE   byWithoutRecv;
    BYTE   byRes[63];
}NET_DVR_LOCAL_PTZ_CFG, *LPNET_DVR_LOCAL_PTZ_CFG;
```

## Members

### **byWithoutRecv**

Whether to receive the response from device: 0-yes, 1-no.

### **byRes**

Reserved, set to 0

## Remarks

This configuration is applicable to 3G network.

## A.70 NET\_DVR\_LOCAL\_CHECK\_DEV

Heartbeat time interval configuration structure.

### Structure Definition

```
struct{
    DWORD      dwCheckOnlineTimeout;
    DWORD      dwCheckOnlineNetFailMax;
    BYTE       byRes[256];
}NET_DVR_LOCAL_CHECK_DEV, *LPNET_DVR_LOCAL_CHECK_DEV;
```

### Members

#### dwCheckOnlineTimeout

Online health monitoring time interval, unit: ms, range: 30-120 (s), 0-120s (default), the recommended value is 30s.

#### dwCheckOnlineNetFailMax

The maximum number of network failure attempts, if the failure attempts are larger than this threshold, exception message will be called back. 0-1 (default), the recommended value is 3.

#### byRes

Reserved, set to 0.

## A.71 NET\_DVR\_LOCAL\_GENERAL\_CFG

General configurations structure.

### Structure Definition

```
struct{
    BYTE      byExceptionCbDirectly;
    BYTE      byNotSplitRecordFile;
    BYTE      byResumeUpgradeEnable;
    BYTE      byAlarmJsonPictureSeparate;
    BYTE      byRes[4];
    UINT64   i64FileSize;
    DWORD     dwResumeUpgradeTimeout;
    BYTE      byAlarmReconnectMode;
    BYTE      byStdXmlBufferSize;
    BYTE      byMultiplexing;
    BYTE      byFastUpgrade;
    BYTE      byRes[232];
}NET_DVR_LOCAL_GENERAL_CFG, *LPNET_DVR_LOCAL_GENERAL_CFG;
```

### Members

#### **byExceptionCbDirectly**

Exception callback type: 0-callback via thread pool, 1-callback via upper-layer.

#### **byNotSplitRecordFile**

Whether to subpackage the local video files: 0-yes (default), 1-no.

#### **byResumeUpgradeEnable**

Whether to enable upgrading ANR (Automatic Network Replenishment): 0-disable (default), 1-enable.

#### **byAlarmJsonPictureSeparate**

Whether to separate the alarm data and the alarm picture which will be transmitted in JSON format: 0-not separate, 1-separate (the **ICommand** in the callback function will be "COMM\_ISAPI\_ALARM").

#### **byRes**

Reserved.

#### **i64FileSize**

Maximum file size, unit: byte. When subpackaging is enabled, if the saved video file size is larger than the value of this parameter, the file will be subpackaged to multiple file segments for storage.

#### **dwResumeUpgradeTimeout**

ANR reconnection timeout, unit: millisecond.

#### **byAlarmReconnectMode**

Reconnection mode: 0-dependent thread reconnection (default), 1-thread pool reconnection.

#### **byStdXmlBufferSize**

Buffer size for receiving data transmitted by ISAPI: 1-1 MB, other values-default.

#### **byMultiplexing**

Whether to enable multiplexing of normal link (non-TLS link): 0-disable, 1-enable.

#### **byFastUpgrade**

Upgrading mode: 1-normal upgrading, 2-fast upgrading.

#### **byRes1**

Reserved.

## A.72 NET\_DVR\_LOCAL\_LOG\_CFG

Log configuration structure.

## Structure Definition

```
struct{
    WORD      wSDKLogNum;
    BYTE      byRes [254];
}NET_DVR_LOCAL_LOG_CFG, *LPNET_DVR_LOCAL_LOG_CFG;
```

## Members

### wSDKLogNum

Number of log files in overwritten mode, "0"-10 log files (default).

### byRes

Reserved, set to 0.

## A.73 NET\_DVR\_LOCAL\_MEM\_POOL\_CFG

Local configuration structure of storage pool.

## Structure Definition

```
struct{
    DWORD      dwAlarmMaxBlockNum;
    DWORD      dwAlarmReleaseInterval;
    BYTE      byRes [60];
}NET_DVR_LOCAL_MEM_POOL_CFG, *LPNET_DVR_LOCAL_MEM_POOL_CFG;
```

## Members

### dwAlarmMaxBlockNum

The maximum number of memory blocks can be applied, the maximum size of each applied block is 64MB, if the required memory block size is larger than the threshold, do not apply for it from the system. If the value of this parameter is set to 0, it refers that the number of memory block can be applied is not limited.

### dwAlarmReleaseInterval

The time interval between each free memory blocks to be released, unit: s, 0-not release the free memory.

### byRes

Reserved, set to 0.

## A.74 NET\_DVR\_LOCAL\_MODULE\_RECV\_TIMEOUT\_CFG

Structure about timeout configuration by module.

## Structure Definition

```
struct{
    DWORD    dwPreviewTime;
    DWORD    dwAlarmTime;
    DWORD    dwVodTime;
    DWORD    dwElse;
    BYTE     byRes[512];
}NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG,
*LPNET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG;
```

## Members

### **dwPreviewTime**

Live view module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

### **dwAlarmTime**

Alarm module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

### **dwVodTime**

Playback module receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

### **dwElse**

Other modules' receiving timeout, unit: millisecond, range: 0-3000,000, 0-restore to default settings.

### **byRes**

Reserved, set to 0.

## A.75 NET\_DVR\_LOCAL\_PORT\_MULTI\_CFG

Configuration parameter structure of port multiplier.

## Structure Definition

```
struct{
    BOOL      bEnable;
    BYTE     byRes[60];
}NET_DVR_LOCAL_PORT_MULTI_CFG, *LPNET_DVR_LOCAL_PORT_MULTI_CFG;
```

## Members

### **bEnable**

Whether to enable port multiplier: true-yes.

**byRes**

Reserved, set to 0.

**See Also**

[NET\\_SDK\\_LOCAL\\_CFG\\_TYPE](#)

## A.76 NET\_DVR\_LOCAL\_PROTECT\_KEY\_CFG

**Key Parameter Structure**

Member	Data Type	Description
byProtectKey	Array of BYTE	Key, the default value is 0. The maximum size is 128 bytes.
byRes	Array of BYTE	Reserved, set to 0. The maximum size is 128 bytes.

## A.77 NET\_DVR\_LOCAL\_SDK\_PATH

**Path Information Structure for Loading Component Libraries**

Member	Data Type	Description
sPath	Array of char	Component libraries' addresses
byRes	Array of BYTE	Reserved.

**Remarks**

If the path of HCNetSDKCom folder and HCNetSDK libraries are same, but the path of executable programs are different, you can call [NET\\_DVR\\_SetSDKInitCfg](#) to specify the path of HCNetSDKCom folder to make sure the component libraries are loaded normally.

## A.78 NET\_DVR\_LOCAL\_STREAM\_CALLBACK\_CFG

## Key Parameter Structure

Member	Data Type	Description
<b>byPlayBackEndFlag</b>	BYTE	Whether to call back playback end flag:0-No, 1-Yes
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 255 bytes.

## A.79 NET\_DVR\_LOCAL\_TALK\_MODE\_CFG

Two-way audio configuration structure.

### Structure Definition

```
struct{
    BYTE    byTalkMode;
    BYTE    byRes[127];
}NET_DVR_LOCAL_TALK_MODE_CFG, *LPNET_DVR_LOCAL_TALK_MODE_CFG;
```

### Members

#### byTalkMode

Two-way audio mode: 0-enable two-way audio library (default), 1-enable Windows API mode.

#### byRes

Reserved, set to 0.

### Remarks

If the two-way audio library is enabled, you must load the "AudioIntercom.dll" and "OpenAL32.dll".

## A.80 NET\_DVR\_LOCAL\_TCP\_PORT\_BIND\_CFG

Local binding configuration structure of TCP port.

### Structure Definition

```
struct{
    WORD      wLocalBindTcpMinPort;
    WORD      wLocalBindTcpMaxPort;
    BYTE      byRes[60];
}NET_DVR_LOCAL_TCP_PORT_BIND_CFG, *LPNET_DVR_LOCAL_TCP_PORT_BIND_CFG;
```

## Members

### wLocalBindTcpMinPort

The minimum TCP port number to be bound locally.

### wLocalBindTcpMaxPort

The maximum TCP port number to be bound locally.

### byRes

Reserved, set to 0.

## Remarks

- Port bind strategy: provide a port number segment to ensure all used port numbers are in the segment (except multicast); the ports from port pool are tried to bind one by one until the port is not occupied, if all ports are occupied, error will be returned; binding the system reserved ports (form 1 to 1024) is not suggested.
- The maximum port number to be bound should be equal to or larger than the minimum port number, [0,0]: clear the binding; [0,non-0]: setting failed, as 0 can't be bound.

## A.81 NET\_DVR\_LOCAL\_UDP\_PORT\_BIND\_CFG

Local binding configuration structure of UDP port.

## Structure Definition

```
struct{
    WORD      wLocalBindUdpMinPort;
    WORD      wLocalBindUdpMaxPort;
    BYTE     byRes[60];
}NET_DVR_LOCAL_UDP_PORT_BIND_CFG, *LPNET_DVR_LOCAL_UDP_PORT_BIND_CFG;
```

## Members

### wLocalBindUdpMinPort

The minimum UDP port number to be bound locally.

### wLocalBindUdpMaxPort

The maximum UDP port number to be bound locally.

### byRes

Reserved, set to 0.

## Remarks

- Port bind strategy: provide a port number segment to ensure all used port numbers are in the segment (except multicast); the ports from port pool are tried to bind one by one until the port

- is not occupied, if all ports are occupied, error will be returned; binding the system reserved ports (form 1 to 1024) is not suggested.
- The maximum port number to be bound should be equal to or larger than the minimum port number, [0,0]: clear the binding; [0,non-0]: setting failed, as 0 can't be bound.

## A.82 NET\_DVR\_MESSAGE\_CALLBACK\_PARAM\_V51

Alarm Callback Configuration Parameters

### Key Parameter Structure

Member	Data Type	Description
<b>byVcaAlarmJsonType</b>	BYTE	JSON format for alarm transmission (COMM_VCA_ALARM): 0-new JSON format, 1-old JSON format.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 63 bytes.

## A.83 NET\_DVR\_MIME\_UNIT

### Input Content Details Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)

Member	Data Type	Description
<b>szContentType</b>	Array of char	Content type (corresponds to <b>Content-Type</b> field in the message), e.g., text/json. text/xml, and so on. The content format must be supported by HTTP.
<b>szName</b>	Array of char	Content name (corresponds to <b>name</b> field in the message), e.g., name="upload".
<b>szFilename</b>	Array of char	Content file name (corresponds to <b>filename</b> field in the message), e.g., filename="C:\Users\test\Desktop\11.txt".
<b>dwContentLen</b>	DWORD	Content size
<b>pContent</b>	char*	Data point

Member	Data Type	Description
bySelfRead	BYTE	0-External file, 1-Internal data, whose address is specified by <b>szFilename</b> .
byRes	Array of BYTE	Reserved. Set to 0. Maximum: 15 bytes.

## See Also

[NET\\_DVR\\_XML\\_CONFIG\\_INPUT](#)

## A.84 NET\_DVR\_MULTI\_CARD\_CFG\_V50

Multi-factor authentication parameter structure.

### Structure Definition

```
struct{
    DWORD                      dwSize;
    BYTE                       byEnable;
    BYTE                       bySwipeIntervalTimeout;
    BYTE                       byRes1[2];
    struGroupCfg[NET_SDK_MULTI_CARD_GROUP_NUM/
*20*];
    BYTE                       byRes2[32];
}NET_DVR_MULTI_CARD_CFG_V50,*LPNET_DVR_MULTI_CARD_CFG_V50;
```

### Members

#### dwSize

Structure size

#### byEnable

Whether to enable multi-factor authentication: 0-no, 1-yes.

#### bySwipeIntervalTimeout

Card swiping interval timeout, which is ranging from 1 to 255, unit: second, default: 10s.

#### byRes1

Reserved, set to 0.

#### struGroupCfg

Card swiping parameters of group, see details in the structure

[NET\\_DVR\\_MULTI\\_CARD\\_GROUP\\_CFG\\_V50](#).

#### byRes2

Reserved, set to 0.

## A.85 NET\_DVR\_MULTI\_CARD\_GROUP\_CFG\_V50

Card swiping parameter structure of card group.

### Structure Definition

```
struct{
    BYTE                                byEnable;
    BYTE                                byEnableOfflineVerifyMode;
    BYTE                                byRes1[2];
    DWORD                               dwTemplateNo;
    NET_DVR_GROUP_COMBINATION_INFO_V50 struGroupCombination[GROUP_COMBINATION_NUM];
}NET_DVR_MULTI_CARD_GROUP_CFG_V50,*LPNET_DVR_MULTI_CARD_GROUP_CFG_V50;
```

### Members

#### byEnable

Whether to enable card group parameters of multi-factor authentication: 0-no, 1-yes.

#### bySwipeIntervalTimeout

Whether to enable access authentication when the access controller is offline (open door by super password instead of remotely opening door): 0-no, 1-yes

#### byRes1

Reserved, set to 0.

#### dwTemplateNo

Template No. of multi-factor authentication schedule, which reuses the template of access permission control schedule.

#### struGroupCombination

Group parameters, see details in the structure [NET\\_DVR\\_GROUP\\_COMBINATION\\_INFO\\_V50](#).

### See Also

[NET\\_DVR\\_MULTI\\_CARD\\_CFG\\_V50](#)

## A.86 NET\_DVR\_NETCFG\_V50

## Network Configuration Structure

Member	Data Type	Description
dwSize	DWORD	Structure size.
struEtherNet	Array of <b><u>NET_DVR_ETHERNET_V30</u></b>	Ethernet interface
struRes1	Array of	Reserved, set to 0.
struAlarmHostIpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	Listening service IP address
byRes2	Array of BYTE	Reserved, set as 0
wAlarmHostIpPort	WORD	Listening service port No.
byUseDhcp	BYTE	Whether to enable DHCP: 0xff- invalid; 0-disable, 1-enable
byIPv6Mode	BYTE	Allocation mode of IPv6 address: 0-by router advertisement, 1-by manual setting, 2-by enabling DHCP allocation.
struDnsServer1IpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	IP address of domain name server 1
struDnsServer2IpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	IP address of domain name server 2
byIpResolver	Array of BYTE	IP resolver domain name or IP address (if the port No. of device is 8000, the domain name is not supported).
wIpResolverPort	WORD	IP resolver port No.
wHttpPortNo	WORD	HTTP port No.
struMulticastIpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	Multicast group address
struGatewayIpAddr	<b><u>NET_DVR_IPADDR_UNION</u></b>	Gateway address
struPPPoE	<b><u>NET_DVR_PPPOECFG</u></b>	PPPoE parameters
byEnablePrivateMulticastDiscovery	BYTE	Private multicast search (SADP): 0-default, 1-enable, 2-disable

Member	Data Type	Description
byEnableOnvifMulticastDiscovery	BYTE	Onvif multicast search (SADP): 0-default, 1-enable, 2-disable
wAlarmHost2IpPort	WORD	Port No. of listening host 2.
struAlarmHost2IpAddr	<u>NET_DVR_IPADDR_UNION</u>	IP address of listening host 2
byEnableDNS	BYTE	DNS address setting mode: 0-automatically get, 1-manually set.
byRes	Array of BYTE	Reserved, set to 0

## Remarks

- For device only supports the private protocol with version 3.0 or lower, when the parameter **byUseDhcp="0xff"**, you should set the device IP address to null, and then the device will automatically get the DHCP information.
- When the parameter **byIpv6Mode** is set to 0 or 2, setting IPv6 address in the parameter **struEtherNet** is not required, it will be obtained automatically by the device; when **byIpv6Mode** is set to 1, you should set IPv6 address. As there are multiple IPv6 addresses, the IPv6 address of current logged-in device may be different with that in **struEtherNet**.

## A.87 NET\_DVR\_PLAN\_TEMPLATE

Schedule template configuration structure.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byEnable;
    BYTE       byRes1[3];
    BYTE       byTemplateName[TEMPLATE_NAME_LEN/*32*/];
    DWORD      dwWeekPlanNo;
    DWORD      dwHolidayGroupNo[MAX_HOLIDAY_GROUP_NUM/*16*/];
    BYTE       byRes2[32];
}NET_DVR_PLAN_TEMPLATE,*LPNET_DVR_PLAN_TEMPLATE;
```

### Members

#### dwSize

Structure size.

#### byEnable

Whether to enable: 1-enable, 0-disable.

**byRes1**

Reserved, set to 0.

**byGroupName**

Schedule template name.

**byGroupName**

Week schedule No.: 0-invalid.

**dwHolidayGroupNo**

Holiday group No.: 0-invalid.

**byRes2**

Reserved, set to 0.

## A.88 NET\_DVR\_PPPOECONFIGURATION

### PPPoE Configuration Structure

Member	Data Type	Description
dwPPPOE	DWORD	Whether to enable PPPoE: 0-no, 1-yes.
sPPPoEUser	Array of BYTE	PPPoE user name.
sPPPoEPASSWORD	Array of char	PPPoE password.
struPPPoEIP	<u>NET_DVR_IPADDR_UNION</u>	PPPoE IP address

## A.89 NET\_DVR\_RECORD\_PASSBACK\_MANUAL\_COND

### Structure About Conditions of Getting Task of Manually Copying Back Videos

Member	Data Type	Description
dwSize	DWORD	Structure size.
byType	BYTE	Method of getting the task information: 0 (get remaining tasks), 1 (get remaining tasks by stream ID), 2 (get all tasks), 3 (get all tasks by stream ID).

Member	Data Type	Description
<b>byRes1</b>	BYTE	Reserved, set to 0. The size is 3 bytes.
<b>struStreamInfo</b>	<u><a href="#">NET_DVR_STREAM_IN</a></u> <u><a href="#">FO</a></u>	Stream information structure. This member is valid when getting the task information by stream ID.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The size is 128 bytes.

## A.90 NET\_DVR\_RECORD\_PASSBACK\_MANUAL\_TASK\_RET

### Structure About Results of Getting Task of Manually Copying Back Videos

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>struStreamInfo</b>	<u><a href="#">NET_DVR_STREAM_IN</a></u> <u><a href="#">FO</a></u>	Stream information structure. This member is valid when getting the task information by stream ID.
<b>dwTaskID</b>	DWORD	Task ID
<b>struStartTime</b>	<u><a href="#">NET_DVR_TIME_EX</a></u>	Start time of video copy-back
<b>struStopTime</b>	<u><a href="#">NET_DVR_TIME_EX</a></u>	End time of video copy back
<b>byTaskStatus</b>	BYTE	Task status: 0 (not executed), 1 (pausing), 2 (executed), 3 (copying back), 4 (copy-back failed), 5 (succeeded, but only some videos are copied back), 6 (succeeded, but there is no video in the camera).
<b>byRes1</b>	Array of BYTE	Reserved, set to 0. The size is 3 bytes.
<b>struExecuteStartTime</b>	<u><a href="#">NET_DVR_TIME_EX</a></u>	Actual start time of executing the task. This member is valid when the value of <b>byTaskStatus</b> is 1 or 2.
<b>struExecuteStopTime</b>	<u><a href="#">NET_DVR_TIME_EX</a></u>	Actual end time of executing the task. This member is valid when the value of <b>byTaskStatus</b> is 1 or 2.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The size is 128 bytes.

## A.91 NET\_DVR\_RTSP\_PARAMS\_CFG

### RTSP Parameter Structure

Member	Data Type	Description
<b>dwMaxBuffRoomNum</b>	DWORD	Maximum number of buffers for RTP over UDP sorting, the default value is 20. If the value is 0, it indicates that the member is invalid. One buffer size is about 10 KB, more number of buffers indicates higher sorting ability, more fluent, and longer delay.
<b>byUseSort</b>	BYTE	Whether to enable RTP over UDP sorting: 0-no, 1-yes.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 123 bytes.

## A.92 NET\_DVR\_SETUPALARM\_PARAM\_V50

### Arming Parameter Structure

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>byLevel</b>	BYTE	Arming priority: 0-high, 1-medium, 2-low.
<b>byAlarmInfoType</b>	BYTE	Intelligent traffic alarm information type: 0-old (NET_DVR_PLATE_RESULT),1-new (NET_ITS_PLATE_RESULT).
<b>byRetAlarmTypeV40</b>	BYTE	0-the motion detection, video loss, video tampering, and alarm input alarm information is uploaded in normal mode (alarm type: COMM_ALARM_V30, alarm information structure: <b>NET_DVR_ALARMINFO_V30</b> ); 1-alarm information is uploaded in variable size (alarm type: COMM_ALARM_V40, alarm information structure: <b>NET_DVR_ALARMINFO_V40</b> ).

Member	Data Type	Description
<b>byRetDevInfoVersion</b>	BYTE	Alarm types of CVR: 0-COMM_ALARM_DEVICE (alarm information structure: <b><i>NET_DVR_ALARMINFO_DEV</i></b> ), 1-COMM_ALARM_DEVICE_V40 (alarm information structure: <b><i>NET_DVR_ALARMINFO_DEV_V40</i></b> ).
<b>byRetVQDAlarmType</b>	BYTE	VQD alarm types: 0-COMM_ALARM_VQD (alarm information structure: <b><i>NET_DVR_VQD_DIAGNOSE_INFO</i></b> ), 1-COMM_ALARM_VQD_EX (alarm information structure: <b><i>NET_DVR_VQD_ALARM</i></b> , including camera information and captured pictures)
<b>byFaceAlarmDetection</b>	BYTE	Face detection alarm types: 1-face detection alarm (alarm type: COMM_ALARM_FACE_DETECTION, alarm information structure: <b><i>NET_DVR_FACE_DETECTION</i></b> ), 0-face capture alarm (alarm type: COMM_UPLOAD_FACESNAP_RESULT, alarm information structure: <b><i>NET_VCA_FACESNAP_RESULT</i></b> ).
<b>bySupport</b>	BYTE	Capabilities, which is represented by bit: <ul style="list-style-type: none"> <li>• bit0-whether to upload picture: 0-yes, 1-no</li> <li>• bit1-whether to enable ANR: 0-no, 1-yes</li> <li>• bit4-whether to upload behavior analysis events of all detection targets: 0-no, 1-yes. It is used to enable the NVR to get events of all targets detected by network cameras.</li> <li>• bit5-whether to enable all-day event or alarm uploading: 0-no, 1-yes. It is used to enable the NVR to receive all alarms from network cameras.</li> </ul>
<b>byBrokenNetHttp</b>	BYTE	ANR type, which is represented by bit and should be supported by device: <ul style="list-style-type: none"> <li>• bit0-whether to enable ANR for ANPR: 0-no, 1-yes.</li> <li>• bit1-whether to enable ANR for people counting: 0-no, 1-yes.</li> <li>• bit2-whether to enable ANR for heat map: 0-no, 1-yes.</li> </ul>

Member	Data Type	Description
		<ul style="list-style-type: none"> <li>bit3-whether to enable ANR for face capture: 0-no, 1-yes.</li> <li>bit4-whether to enable ANR for face picture comparison: 0-no, 1-yes.</li> <li>bit5-whether to enable ANR for JSON message transmission: 0-no, 1-yes.</li> <li>bit6: whether to enable ANR for uploading heat map data by dwell time duration and by people quantity: 0-no, 1-yes.</li> <li>bit7: whether to enable ANR for uploading intersection analysis result: 0-no, 1-yes.</li> </ul>
wTaskNo	BYTE	Task No.
byDeployType	BYTE	Arming type: 0-arm via client software, 1-real-time arming.
bySubSCRIPTION	BYTE	<p>Subscription parameters, which is represent by bit.</p> <p>Bit7-whether to upload picture after subscribing motion detection alarm by person or vehicle: 0-no, 1-yes.</p>
byRes1	Array [BYTE]	Reserved, set to 0. The maximum size is 2 bytes.
byAlarmTypeURL	BYTE	<p>Alarm picture data type, which is represented by bit, if the device supports uploading alarm pictures in binary format and URL format, you can specify the data type to be uploading via this parameter, if the device only supports URL format, this parameter is invalid. If the URL format is selected, you should set the device and enable the cloud storage, otherwise, the picture will still be transmitted in binary format.</p> <ul style="list-style-type: none"> <li>bit0-type of captured face pictures: 0-binary data, 1-URL</li> <li>bit1-type of picture uploaded in message: 0-binary, 1-URL</li> <li>bit2-type of picture uploaded for face picture comparison: 0-binary, 1-URL</li> </ul>

Member	Data Type	Description
<b>byCustomCtrl</b>	BYTE	Custom control type, which is represented by bit, bit0-whether to upload the face thumbnail of the front passenger: 0-no, 1-yes
<b>byRes4</b>	Array [BYTE]	Reserved, set to 0. The maximum size is 128 bytes.

## Remarks

- The parameters **byLevel** and **byAlarmInfoType** are available for traffic cameras. Up to 1 cameras can be armed in the priority of level 0, up to 3 cameras can be armed in the priority of level 1, and up to 5 cameras can be armed in the priority of level 3, the alarm/event information from the camera in highest priority will be uploaded first.
- For arming via client software, only supports arming one channel, and supports uploading the alarm/event when device is offline; for real-time arming, up to four channels can be armed at same time, but uploading alarm/event when device is offline is not supported.
- The parameter **wTaskNo** is used to distinguish different arming connections. If the value of this parameter in different arming connections is same, error will be returned.

## A.93 NET\_DVR\_SIMPLE\_DAYTIME

Time parameter structure.

### Structure Definition

```
struct{
    BYTE    byHour;
    BYTE    byMinute;
    BYTE    bySecond;
    BYTE    byRes;
}NET_DVR_SIMPLE_DAYTIME,*LPNET_DVR_SIMPLE_DAYTIME;
```

### Members

#### **byHour**

Hour

#### **byMinute**

Minute

#### **bySecond**

Second

#### **byRes**

Reserved, set to 0.

## A.94 NET\_DVR\_SIMXML\_LOGIN

### Structure about Complement Fields by Stimulation Capability

Member	Data Type	Description
<b>byLoginWithSimXml</b>	BYTE	Whether to complement fields by stimulation capability: 0-no, 1-yes.
<b>byRes</b>	Array of BYTE	Reserved, set to 0. The maximum size is 127 bytes.

## A.95 NET\_DVR\_SINGLE\_PLAN\_SEGMENT

Parameter structure of a schedule

### Structure Definition

```
struct{
    BYTE           byEnable;
    BYTE           byDoorStatus;
    BYTE           byVerifyMode;
    BYTE           byRes[5];
    NET_DVR_TIME_SEGMENT struTimeSegment;
}NET_DVR_SINGLE_PLAN_SEGMENT, *LPNET_DVR_SINGLE_PLAN_SEGMENT;
```

### Members

#### byEnable

Whether to enable: 1-no, 0-yes.

#### byDoorStatus

Door status: 0-invalid, 1-remain open (access without authentication), 2-remain closed (access is not allowed), 3-normal (access by authentication).

#### byVerifyMode

Authentication mode: 0-invalid, 1-card, 2-card+password, 3-password, 4-card or password, 5-fingerprint, 6-fingerprint+password, 7-fingerprint or password, 8-fingerprint+card, 9-fingerprint+card+password, 10-face or fingerprint or card or password, 11-face+fingerprint, 12-face+password, 13-face+card, 14-face, 15-employee ID+password, 16-fingerprint or password, 17-employee ID+fingerprint, 18-employee ID+fingerprint+password, 19-face+fingerprint+card, 20-face+password+fingerprint, 21-employee ID+face, 22-face or face+card, 23-fingerprint or face,

24-card or face or password, 25-card or face, 26-card or face or fingerprint, 27-card or fingerprint or password.

### **byRes**

Reserved, set to 0.

### **struTimeSegment**

Time period parameters, see [\*\*NET\\_DVR\\_TIME\\_SEGMENT\*\*](#) for details.

## A.96 NET\_DVR\_STREAM\_INFO

Stream information structure.

### Structure Definition

```
struct{
    DWORD      dwSize;
    BYTE       byID [STREAM_ID_LEN/*32*/];
    DWORD      dwChannel;
    BYTE       byRes [32];
}NET_DVR_STREAM_INFO, *LPNET_DVR_STREAM_INFO;
```

### Members

#### **dwSize**

Structure size.

#### **byID**

Stream ID, which consists of letters, digits, and dashes, 0-invalid.

#### **dwChannel**

Linked device channel. When it is 0xffffffff, if setting the stream source, this parameter indicates that no device channel is linked; if setting configuration condition, this parameter is invalid.

#### **byRes**

Reserved, set to 0.

### Remarks

- If the device does not support marking stream ID, e.g., DVR, the parameter **byID** should be set to 0.
- For transcoder, when setting the stream source, only one of **byID** and **dwChannel** can be valid; when transcoding, both the **byID** and **dwChannel** can be invalid, the transcoding channel or stream ID is automatically allocated by device.
- For other devices (e.g., CVR), when this structure is inputted as configuration condition, if both the **byID** and **dwChannel** are invalid, error code (17) will be returned, if they are valid, but

mismatched, error may also be returned, so only setting one of these two parameters is suggested.

## A.97 NET\_DVR\_TIME

### Time Parameter Structure

Member	Data Type	Description
dwYear	DWORD	Year
dwMonth	DWORD	Month
dwDay	DWORD	Day
dwHour	DWORD	Hour
dwMinute	DWORD	Minute
dwSecond	DWORD	Second

## A.98 NET\_DVR\_TIME\_EX

### Extended Time Parameter Structure

Member	Data Type	Description
wYear	WORD	Year
byMonth	BYTE	Month
byDay	BYTE	Day
byHour	BYTE	Hour
byMinute	BYTE	Minute
bySecond	BYTE	Second
byRes	BYTE	Reserved.

## A.99 NET\_DVR\_TIME\_SEGMENT

Time period parameter structure.

## Structure Definition

```
struct{
    NET_DVR_SIMPLE_DAYTIME struBeginTime;
    NET_DVR_SIMPLE_DAYTIME struEndTime;
}NET_DVR_TIME_SEGMENT, *LPNET_DVR_TIME_SEGMENT;
```

## Members

### struBeginTime

Start time of time period, refer to the structure [NET\\_DVR\\_SIMPLE\\_DAYTIME](#) for details.

### struEndTime

End time of time period, refer to the structure [NET\\_DVR\\_SIMPLE\\_DAYTIME](#) for details.

## A.100 NET\_DVR\_TIME\_V30

### Time Parameter Structure

Member	Data Type	Description
wYear	WORD	Year.
byMonth	BYTE	Month.
byDay	BYTE	Day.
byHour	BYTE	Hour.
byMinute	BYTE	Minute.
bySecond	BYTE	Second.
byISO8601	BYTE	Whether the time is in ISO8601 format, i.e., whether the time difference is valid. 0-invalid, the time is device local time, 1-valid.
wMilliSec	WORD	Millisecond.
cTimeDifferenceH	char	Time difference between time and UTC time, unit: hour, the value is between -12 and +14 ("+" indicates the east time

Member	Data Type	Description
		zone), it is valid when <b>byISO8601</b> is "1".
cTimeDifferenceM	char	Time difference between time and UTC time, unit: minute, the value is -30, +30, or +45 ("+" indicates the east time zone), it is valid when <b>byISO8601</b> is "1".

## A.101 NET\_DVR\_USER\_LOGIN\_INFO

### Structure About Login Parameters

Member	Data Type	Description
sDeviceAddress	char	Device IP address, or domain name.
byUseTransport	BYTE	Enable capability transmission or not: 0-no (default), 1-yes.
wPort	WORD	Device port number, e.g., 8000 (when login by private protocol), 80 (when login by text protocol).
sUserName	char	User name for logging in to device.
sPassword	char	Login password.
cbLoginResult	<i>fLoginResultCallBack</i>	Callback function used to return login status, it is valid only when <b>bUseAsynLogin</b> is "1".
pUser	void*	User data.
bUseAsynLogin	BOOL	Whether to enable asynchronous login: 0-no, 1-yes.
byProxyType	BYTE	Proxy server type: 0-no proxy, 1-standard proxy, 2-EHome proxy.
byUseUTCTime	BYTE	0-not convert (default), 1-input or output UTC time, 2-input or output local time.
byLoginMode	BYTE	Login mode: 0-login by private protocol, 1-login by text protocol, 2-self-adaptive (it is available when the protocol type supported by device is

Member	Data Type	Description
		unknown, and this mode does not support asynchronous login).
byHttps	BYTE	Whether to enable TLS for login (by private protocol or by text protocol): 0-no, 1-yes, 2-self-adaptive (which is usually used when the protocol type supported by device is unknown. Both HTTP and HTTPS requests will be sent).
iProxyID	LONG	Proxy server No.
byVerifyMode	BYTE	Whether to enable verification mode: 0-no, 1-bidirectional verification (currently not available), 2-unidirectional verification (it is valid when <b>byLoginMode</b> is 0 and <b>byHttps</b> is 1); when <b>byVerifyMode</b> is 0, CA certificate is not required, when <b>byVerifyMode</b> is 2, you should call NET_DVR_SetSDKLocalCfg to load CA certificate, and the enumeration value is "NET_SDK_LOCAL_CFG_CERTIFICATION".
byRes3	BYTE[]	Reserved, the maximum length is 119 bytes.

## A.102 NET\_DVR\_VALID\_PERIOD\_CFG

Expiry date configuration structure.

### Structure Definition

```
struct{
    BYTE          byEnable;
    BYTE          byBeginTimeFlag;
    BYTE          byEnableTimeFlag;
    BYTE          byTimeDurationNo;
    NET_DVR_TIME_EX struBeginTime;
    NET_DVR_TIME_EX struEndTime;
    BYTE          byTimeType;
    BYTE          byRes2[32];
}NET_DVR_VALID_PERIOD_CFG, *LPNET_DVR_VALID_PERIOD_CFG;
```

### Members

#### byEnable

Whether to enable the expiry date: 0-no, 1-yes.

### **byBeginTimeFlag**

Whether to enable the flag to limit the start time: 0-no, 1-yes.

### **byEnableTimeFlag**

Whether to enable the flag to limit the end time: 0-no, 1-yes.

### **byTimeDurationNo**

Expiry date index No., which starts from 0.

### **struBeginTime**

Start time of the expiry date, see details in the structure [NET\\_DVR\\_TIME\\_EX](#).

### **struEndTime**

End time of the expiry date, see details in the structure [NET\\_DVR\\_TIME\\_EX](#) ..

### **byTimeType**

Time type: 0-device's local time (default), 1-UTC time. This member is valid for **struBeginTime** and **struEndTime**.

### **byRes2**

Reserved, set to 0.

## See Also

[NET\\_DVR\\_GROUP\\_CFG](#)

## A.103 NET\_DVR\_WEEK\_PLAN\_Cfg

Week schedule parameter structure.

### Structure Definition

```
struct{
    DWORD                      dwSize;
    BYTE                       byEnable;
    BYTE                       byRes1[3];
    NET_DVR_SINGLE_PLAN_SEGMENT struPlanCfg[MAX_DAYS/*7*/][MAX_TIMESEGMENT_V30/
*8*/];
    BYTE                       byRes2[16];
}NET_DVR_WEEK_PLAN_Cfg, *LPNET_DVR_WEEK_PLAN_Cfg;
```

## Members

### **dwSize**

Structure size.

### **byEnable**

Whether to enable: 1-no, 0-yes.

**byRes1**

Reserved, set to 0.

**struPlanCfg**

Week schedule parameters, up to 7 days can be set in one week, and up to 8 time periods can be set in one day, see [NET\\_DVR\\_SINGLE\\_PLAN\\_SEGMENT](#) for details.

**byRes2**

Reserved, set to 0.

## A.104 NET\_DVR\_XML\_CONFIG\_INPUT

### Input Parameter Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)

Member	Data Type	Description
<b>dwSize</b>	DWORD	Structure size.
<b>lpRequestUrl</b>	void*	Request URL (command) for implement different functions, and it is in string format.
<b>dwRequestUrlLen</b>	DWORD	Request URL size.
<b>lpInBuffer</b>	void*	Buffer for storing input parameters (request messages), see the input content details structure in <a href="#"><u>NET_DVR_MIME_UNIT</u></a> .
<b>dwInBufferSize</b>	DWORD	Input buffer size.
<b>dwRecvTimeOut</b>	DWORD	Receiving timeout, unit: ms, 0-5000ms (default).
<b>byForceEncrpt</b>	BYTE	Whether to enable force encryption (the messages will be encrypted by AES algorithm for transmission): 0-no, 1-yes.
<b>byNumOfMultiPart</b>	BYTE	Number of message segments: 0-invalid; other values-number of message segments, which is transmitted by the parameter <b>lpInBuffer</b> in the structure <a href="#"><u>NET_DVR_MIME_UNIT</u></a> .
<b>byRes</b>	Array of BYTE	Reserved, set to 0.

### Related API

#### [NET\\_DVR\\_STDXMLConfig](#)

## A.105 NET\_DVR\_XML\_CONFIG\_OUTPUT

### Output Parameter Structure of Message Transmission API (NET\_DVR\_STDXMLConfig)

Member	Data Type	Description
dwSize	DWORD	Structure size.
lpOutBuffer	void*	Buffer for storing output parameters (response messages), which is allocated when passing through URL by GET method.
dwOutBufferSize	DWORD	Output buffer size.
dwReturnedXMLSize	DWORD	Actual size of response message.
lpStatusBuffer	void*	Response status (ResponseStatus message). This parameter will not be assigned if performing GET operation succeeded, and you can also set it to "NULL" if not required.
dwStatusSize	DWORD	Size of response status buffer.
lpDataBuffer	HPR_VOIDPTR	Buffer for transmitted data. This parameter is valid when the value of <b>byNumOfMultiPart</b> is larger than 0.
byNumOfMultiPart	HPR_UINT8	Number of parts that the message is divided into.
byRes [23]	BYTE	Reserved, set to 0.

#### Related API

[NET\\_DVR\\_STDXMLConfig](#)

## A.106 NET\_SDK\_CALLBACK\_STATUS\_NORMAL

## Enumeration About Persistent Connection Status

Enumeration Type	Marco Definition Value	Description
NET_SDK_CALLBACK_STATUS_SUCCESS	1000	Succeeded.
NET_SDK_CALLBACK_STATUS_PROCESSING	1001	Connecting. The <b>IpBuffer</b> is 4-byte status.
NET_SDK_CALLBACK_STATUS_FAILED	1002	Failed. The <b>IpBuffer</b> is the value of 4-byte status and 4-byte error code.

## A.107 NET\_SDK\_DOWNLOAD\_TYPE

Enumerate file types to be downloaded.

### Enumeration Definition

```
typedef enum {
    NET_SDK_DOWNLOAD_CERT = 0,
    NET_SDK_DOWNLOAD_IPC_CFG_FILE = 1,
    NET_SDK_DOWNLOAD_BASELINE_SCENE_PIC = 2,
    NET_SDK_DOWNLOAD_VQD_ALARM_PIC = 3,
    NET_SDK_DOWNLOAD_CONFIGURATION_FILE = 4,
    NET_SDK_DOWNLOAD_SCENE_CONFIGURATION_FILE = 5,
    NET_SDK_DOWNLOAD_FILE_FORM_DB = 6,
    NET_SDK_DOWNLOAD_TME_FILE = 7,
    NET_SDK_DOWNLOAD_VEHICLE_BLOCKALLOWLIST_FILE = 8,
    NET_SDK_DOWNLOAD_GUID_FILE = 9,
    NET_SDK_DOWNLOAD_FILE_FORM_CLOUD = 10,
    NET_SDK_DOWNLOAD_PICTURE = 11,
    NET_SDK_DOWNLOAD_VIDEO = 12,
    NET_DVR_DOWNLOAD_SCREEN_FILE = 13,
    NET_SDK_DOWNLOAD_PUBLISH_MATERIAL = 14,
    NET_SDK_DOWNLOAD_THERMOMETRIC_FILE = 15,
    NET_SDK_DOWNLOAD_LED_CHECK_FILE = 16,
    NET_SDK_DOWNLOAD_VEHICLE_INFORMATION = 17,
    NET_SDK_DOWNLOAD_CERTIFICATE_BLOCKLIST_TEMPLATE = 18,
    NET_SDK_DOWNLOAD_LOG_FILE = 19,
    NET_SDK_DOWNLOAD_FILEVOLUME_DATA = 20,
    NET_SDK_DOWNLOAD_FD_DATA = 21,
    NET_SDK_DOWNLOAD_SECURITY_CFG_FILE = 22,
    NET_SDK_DOWNLOAD_PUBLISH_SCHEDULE = 23,
    NET_SDK_DOWNLOAD_RIGHT_CONTROLLER_AUDIO = 24,
    NET_SDK_DOWNLOAD_MODBUS_CFG_FILE = 25,
```

NET_SDK_DOWNLOAD_RS485_PROTOCOL_DLL_FILE	= 26,
NET_SDK_DOWNLOAD_CLUSTER_MAINTENANCE_LOG	= 27,
NET_SDK_DOWNLOAD_SQL_ARCHIVE_FILE	= 28,
NET_SDK_DOWNLOAD_SUBWIND_STREAM	= 29,
NET_SDK_DOWNLOAD_DEVTYPE_CALIBFILE	= 30,
NET_SDK_DOWNLOAD_HD_CAMERA_CORRECT_TABLE	= 31,
NET_SDK_DOWNLOAD_CLIENT_CALIBFILE	= 32,
NET_SDK_DOWNLOAD_FOUE_CAMERAS_PICTURES	= 33,
NET_SDK_DOWNLOAD_DOOR_CONTENT	= 34,
NET_SDK_DOWNLOAD_PUBLISH_MATERIAL_THUMBNAIL	= 35,
NET_SDK_DOWNLOAD_PUBLISH_PROGRAM_THUMBNAIL	= 36,
NET_SDK_DOWNLOAD_PUBLISH_TEMPLATE_THUMBNAIL	= 37,
NET_SDK_DOWNLOAD_DARK_FIGHTER_X_CORRECT_TABLE_MAIN	= 38,
NET_SDK_DOWNLOAD_DARK_FIGHTER_X_CORRECT_TABLE_BACKUP	= 39,
NET_SDK_DOWNLOAD_OFFLINE_CAPTURE_INFO_TEMPLATE	= 40,
NET_SDK_DOWNLOAD_CAPTURE_DATA	= 41,
NET_SDK_DOWNLOAD_HD_CAMERA_CORRECT_TABLE_FILE	= 42,
NET_SDK_DOWNLOAD_CLIENT_CALIBFILE_FILE	= 43,
NET_SDK_DOWNLOAD_FOUR_CAMERAS_PICTURES_FILE	= 44,
NET_SDK_DOWNLOAD_SCENE_FILE	= 45,
NET_SDK_DOWNLOAD_OPEN_SOURCE_CERT	= 46,
NET_SDK_DOWNLOAD_RATIOSTITCHING_FILE	= 47,
NET_SDK_DOWNLOAD_LENS_PARAM_FILE	= 48,
NET_SDK_DOWNLOAD_SELECT_DEVTYPE_CALIBFILE	= 49
}	NET_SDK_DOWNLOAD_TYPE;

## Enumeration Type

### **NET\_SDK\_DOWNLOAD\_CERT**

Certificate.

### **NET\_SDK\_DOWNLOAD\_IPC\_CFG\_FILE**

Network camera configuration file.

### **NET\_SDK\_DOWNLOAD\_BASELINE\_SCENE\_PIC**

Base scene picture.

### **NET\_SDK\_DOWNLOAD\_VQD\_ALARM\_PIC**

VQD (video quality diagnosis) alarm picture.

### **NET\_SDK\_DOWNLOAD\_CONFIGURATION\_FILE**

Configuration file.

### **NET\_SDK\_DOWNLOAD\_SCENE\_CONFIGURATION\_FILE**

Scene configuration file.

### **NET\_SDK\_DOWNLOAD\_FILE\_FORM\_DB**

File in the image and video library.

### **NET\_SDK\_DOWNLOAD\_TME\_FILE**

Entrance and exit management file.

**NET\_SDK\_DOWNLOAD\_VEHICLE\_BLOCKALLOWLIST\_FILE**

Blocklist and allowlist configuration file.

**NET\_SDK\_DOWNLOAD\_GUID\_FILE**

GUID file.

**NET\_SDK\_DOWNLOAD\_FILE\_FORM\_CLOUD**

Picture in the cloud storage.

**NET\_SDK\_DOWNLOAD\_PICTURE**

Picture.

**NET\_SDK\_DOWNLOAD\_VIDEO**

Video.

**NET\_DVR\_DOWNLOAD\_SCREEN\_FILE**

Screen server file.

**NET\_SDK\_DOWNLOAD\_PUBLISH\_MATERIAL**

Local material file of information release.

**NET\_SDK\_DOWNLOAD\_THERMOMETRIC\_FILE**

Thermometry calibration file.

**NET\_SDK\_DOWNLOAD\_LED\_CHECK\_FILE**

LED correction file.

**NET\_SDK\_DOWNLOAD\_VEHICLE\_INFORMATION**

Vehicle information to be exported.

**NET\_SDK\_DOWNLOAD\_CERTIFICATE\_BLOCKLIST\_TEMPLET**

ID card blocklist template.

**NET\_SDK\_DOWNLOAD\_LOG\_FILE**

Log to be exported.

**NET\_SDK\_DOWNLOAD\_FILEVOLUME\_DATA**

File volume data file, currently it is only supported by CVR (central video recorder) devices.

**NET\_SDK\_DOWNLOAD\_FD\_DATA**

Data in a specific face picture library to be exported.

**NET\_SDK\_DOWNLOAD\_SECURITY\_CFG\_FILE**

Configuration file to be securely exported.

**NET\_SDK\_DOWNLOAD\_PUBLISH\_SCHEDULE**

Schedule to be exported.

**NET\_SDK\_DOWNLOAD\_RIGHT\_CONTROLLER\_AUDIO**

Audio file of the main controller.

**NET\_SDK\_DOWNLOAD\_MODBUS\_CFG\_FILE**

Configuration file of Modbus protocol.

**NET\_SDK\_DOWNLOAD\_RS485\_PROTOCOL\_DLL\_FILE**

Dynamic library file of RS-485 protocol.

**NET\_SDK\_DOWNLOAD\_CLUSTER\_MAINTENANCE\_LOG**

Cluster maintenance log to be exported.

**NET\_SDK\_DOWNLOAD\_SQL\_ARCHIVE\_FILE**

Archived record in the database to be exported.

**NET\_SDK\_DOWNLOAD\_SUBWIND\_STREAM**

Sub-window stream to be exported.

**NET\_SDK\_DOWNLOAD\_DEVTYPE\_CALIBFILE**

Model calibration file to be exported (\*.cal).

**NET\_SDK\_DOWNLOAD\_HD\_CAMERA\_CORRECT\_TABLE**

24 MP/32 MP correction list to be exported (\*.cal).

**NET\_SDK\_DOWNLOAD\_CLIENT\_CALIBFILE**

Client calibration file to be exported (\*.pto).

**NET\_SDK\_DOWNLOAD\_FOUE\_CAMERAS\_PICTURE**

Four-channel picture package to be exported (.tar).

**NET\_SDK\_DOWNLOAD\_DOOR\_CONTENT**

Door contact information.

**NET\_SDK\_DOWNLOAD\_PUBLISH\_MATERIAL\_THUMBNAIL**

Thumbnail of local information release material.

**NET\_SDK\_DOWNLOAD\_PUBLISH\_PROGRAM\_THUMBNAIL**

Thumbnail of information release program.

**NET\_SDK\_DOWNLOAD\_PUBLISH\_TEMPLATE\_THUMBNAIL**

Thumbnail of information release template.

**NET\_SDK\_DOWNLOAD\_DARK\_FIGHTER\_X\_CORRECT\_TABLE\_MAIN**

DarkfighterX correction list file (main partition).

**NET\_SDK\_DOWNLOAD\_DARK\_FIGHTER\_X\_CORRECT\_TABLE\_BACKUP**

DarkfighterX correction list file (backup partition).

**NET\_SDK\_DOWNLOAD\_OFFLINE\_CAPTURE\_INFO\_TEMPLATE**

User list template of collection.

**NET\_SDK\_DOWNLOAD\_CAPTURE\_DATA**

Offline collected data.

### **NET\_SDK\_DOWNLOAD\_HD\_CAMERA\_CORRECT\_TABLE\_FILE**

HD camera correction sheet (CAL format).

### **NET\_SDK\_DOWNLOAD\_CLIENT\_CALIBFILE\_FILE**

User calibration file (PTO format).

### **NET\_SDK\_DOWNLOAD\_FOUR\_CAMERAS\_PICTURES\_FILE**

Channel pictures package (TAR format).

### **NET\_SDK\_DOWNLOAD\_SCENE\_FILE**

Scene file.

### **NET\_SDK\_DOWNLOAD\_OPEN\_SOURCE\_CERT**

Open source license compliance.

### **NET\_SDK\_DOWNLOAD\_RATIOSTITCHING\_FILE**

Ratio stitching file.

### **NET\_SDK\_DOWNLOAD\_LENS\_PARAM\_FILE**

Lens parameters file.

### **NET\_SDK\_DOWNLOAD\_SELECT\_DEVTYPE\_CALIBFILE**

Calibration file in CAL format.

## **A.108 NET\_SDK\_LOCAL\_CFG\_TYPE**

Enumerate the local configuration types of device network SDK.

### **Enumeration Definition**

```
enum {
    NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND          =0,
    NET_SDK_LOCAL_CFG_TYPE_UDP_PORT_BIND          =1,
    NET_SDK_LOCAL_CFG_TYPE_MEM_POOL               =2,
    NET_SDK_LOCAL_CFG_TYPE_MODULE_RECV_TIMEOUT   =3,
    NET_SDK_LOCAL_CFG_TYPE_ABILITY_PARSE         =4,
    NET_SDK_LOCAL_CFG_TYPE_TALK_MODE              =5,
    NET_SDK_LOCAL_CFG_TYPE_PROTECT_KEY            =6
    NET_SDK_LOCAL_CFG_TYPE_CFG_VERSION           =7
    NET_SDK_LOCAL_CFG_TYPE_RTSP_PARAMS            =8
    NET_SDK_LOCAL_CFG_TYPE_SIMXML_LOGIN          =9
    NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV              =10,
    NET_SDK_LOCAL_CFG_TYPE_SECURITY               =11
    NET_SDK_LOCAL_CFG_TYPE_EZVIZLIB_PATH          =12
    NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE            =13,
    NET_SDK_LOCAL_CFG_TYPE_PROXYSYS              =14
    NET_DVR_LOCAL_CFG_TYPE_LOG                   =15
    NET_DVR_LOCAL_CFG_TYPE_STREAM_CALLBACK       =16
    NET_DVR_LOCAL_CFG_TYPE_GENERAL               =17,
    NET_DVR_LOCAL_CFG_TYPE_PTZ                  =18,
```

```
NET_DVR_LOCAL_CFG_MESSAGE_CALLBACK_V51      =19
NET_SDK_LOCAL_CFG_CERTIFICATION             =20,
NET_SDK_LOCAL_CFG_PORT_MULTIPLEX            =21,
NET_SDK_LOCAL_CFG_ASYNC                     =22
}NET_SDK_LOCAL_CFG_TYPE
```

## Members

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_TCP\_PORT\_BIND**

Local binding configuration of TCP port, see details in [NET\\_DVR\\_LOCAL\\_TCP\\_PORT\\_BIND\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_UDP\_PORT\_BIND**

Binding configuration of local UDP port, see details in  
[NET\\_DVR\\_LOCAL\\_UDP\\_PORT\\_BIND\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_MEM\_POOL**

Local configuration of storage pool, see details in [NET\\_DVR\\_LOCAL\\_MEM\\_POOL\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_MODULE\_RECV\_TIMEOUT**

Timeout configuration by module, see details in  
[NET\\_DVR\\_LOCAL\\_MODULE\\_RECV\\_TIMEOUT\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_ABILITY\_PARSE**

Capability analysis library configuration, see details in [NET\\_DVR\\_LOCAL\\_ABILITY\\_PARSE\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_TALK\_MODE**

Two-way audio configuration, see details in [NET\\_DVR\\_LOCAL\\_TALK\\_MODE\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_PROTECT\_KEY**

Key configuration, see details in [NET\\_DVR\\_LOCAL\\_PROTECT\\_KEY\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_CFG\_VERSION**

Check the device compatibility when setting parameters.

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_RTSP\_PARAMS**

RTSP parameters, see details in [NET\\_DVR\\_RTSP\\_PARAMS\\_CFG](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_SIMXML\_LOGIN**

Parameters of using stimulation capability to complement fields, see details in  
[NET\\_DVR\\_SIMXML\\_LOGIN](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_CHECK\_DEV**

Heartbeat time interval, see details in [NET\\_DVR\\_LOCAL\\_CHECK\\_DEV](#).

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_SECURITY**

SDK security parameters.

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_EZVIZLIB\_PATH**

Communication library address of EZVIZ cloud.

### **NET\_SDK\_LOCAL\_CFG\_TYPE\_CHAR\_ENCODE**

Encoding format conversion configuration, see details in [\*\*NET\\_DVR\\_LOCAL\\_BYTE\\_ENCODE\\_CONVERT\*\*](#).

#### **NET\_SDK\_LOCAL\_CFG\_TYPE\_PROXY**

Proxy types.

#### **NET\_DVR\_LOCAL\_CFG\_TYPE\_LOG**

Log parameters, see details in [\*\*NET\\_DVR\\_LOCAL\\_LOG\\_CFG\*\*](#).

#### **NET\_DVR\_LOCAL\_CFG\_TYPE\_STREAM\_CALLBACK**

Stream callback parameters, see details in [\*\*NET\\_DVR\\_LOCAL\\_STREAM\\_CALLBACK\\_CFG\*\*](#).

#### **NET\_DVR\_LOCAL\_CFG\_TYPE\_GENERAL**

General parameters, see details in [\*\*NET\\_DVR\\_LOCAL\\_GENERAL\\_CFG\*\*](#).

#### **NET\_DVR\_LOCAL\_CFG\_TYPE\_PTZ**

PTZ interaction parameters, see details in [\*\*NET\\_DVR\\_LOCAL\\_CFG\\_TYPE\\_PTZ\*\*](#).

#### **NET\_DVR\_LOCAL\_CFG\_MESSAGE\_CALLBACK\_V51**

Local parameters of alarm callback, see details in [\*\*NET\\_DVR\\_MESSAGE\\_CALLBACK\\_PARAM\\_V51\*\*](#).

#### **NET\_SDK\_LOCAL\_CFG\_CERTIFICATION**

Certificate parameters, see details in [\*\*NET\\_DVR\\_LOCAL\\_CERTIFICATION\*\*](#).

#### **NET\_SDK\_LOCAL\_CFG\_PORT\_MULTIPLICITY**

Port multiplier parameters, see details in [\*\*NET\\_DVR\\_LOCAL\\_PORT\\_MULTI\\_CFG\*\*](#).

#### **NET\_SDK\_LOCAL\_CFG\_ASYNC**

Asynchronous mode parameters, see details in [\*\*NET\\_DVR\\_LOCAL\\_ASYNC\\_CFG\*\*](#).

## **A.109 NET\_SDK\_UPLOAD\_TYPE**

### **Enumeration about File Types to Be Uploaded**

Enumeration Type	Macro Definition Value	Description
<b>UPGRADE_CERT_FILE</b>	0	Certificate file to be upgraded.
<b>UPLOAD_CERT_FILE</b>	1	Certificate file to be uploaded.
<b>TRIAL_CERT_FILE</b>	2	Trial license file.
<b>CONFIGURATION_FILE</b>	3	Configuration file.
<b>UPLOAD_RECORD_FILE</b>	4	Video file.
<b>SCENE_CONFIGURATION_FILE</b>	5	Scene configuration file.

Enumeration Type	Macro Definition Value	Description
<b>UPLOAD_PICTURE_FILE</b>	6	Picture file.
<b>UPLOAD_VIOLATION_FILE</b>	7	Violation dictionary file.
<b>UPLOAD_TG_FIL</b>	8	Timing generator file.
<b>UPLOAD_DATA_TO_DB</b>	9	File to be uploaded to picture and video library.
<b>UPLOAD_BACKGROUND_PIC</b>	10	Background picture.
<b>UPLOAD_CALIBRATION_FILE</b>	11	Calibration file.
<b>UPLOAD_TME_FILE</b>	12	Entrance and exiting management file.
<b>UPLOAD_VEHICLE_BLOCKALLOWLST_FILE</b>	13	Vehicle blocklist file.
<b>UPLOAD_PICTURE_TO_CLOUD</b>	15	Picture file to be uploaded to cloud storage.
<b>UPLOAD_VIDEO_FILE</b>	16	Video file.
<b>UPLOAD_SCREEN_FILE</b>	17	Screen server file.
<b>UPLOAD_PUBLISH_MATERIAL</b>	18	Local material file of information release system.
<b>UPLOAD_PUBLISH_UPGRADE_FILE</b>	19	Upgrade file of information release system.
<b>UPLOAD_RING_FILE</b>	20	Ringtone file.
<b>UPLOAD_ENCRYPT_CERT</b>	21	Encryption certificate.
<b>UPLOAD_THERMOMETRIC_FILE</b>	22	Calibration file for temperature measurement.
<b>UPLOAD_SUBBRAND_FILE</b>	23	Vehicle sub brand file.
<b>UPLOAD_LED_CHECK_FILE</b>	24	LED correction file.
<b>BATCH_UPLOAD_PICTURE_FILE</b>	25	Picture files for uploading in batch.
<b>UPLOAD_EDID_CFG_FILE</b>	26	EDID configuration file.
<b>UPLOAD_PANORAMIC_STITCH</b>	27	Panorama stitching configuration file.
<b>UPLOAD_BINOCULAR_COUNTING</b>	28	Binocular counting correction sheet.

Enumeration Type	Macro Definition Value	Description
<b>UPLOAD_AUDIO_FILE</b>	29	Audio file.
<b>UPLOAD_PUBLISH_THIRD_PARTY_FILE</b>	30	Third-party file.
<b>UPLOAD_DEEPEYES_BINOCULAR</b>	31	TX1 binocular correction sheet.
<b>UPLOAD_CERTIFICATE_BLOCKLIST</b>	32	ID card blocklist.
<b>UPLOAD_HD_CAMERA_CORRECT_TABLE</b>	33	HD camera correction sheet (CAL format).
<b>UPLOAD_FD_DATA</b>	35	Face data file to be imported to face picture library.
<b>UPLOAD_FACE_DATA</b>	36	Face picture file to be imported to face picture library.
<b>UPLOAD_FACE_ANALYSIS_DATA</b>	37	Picture file to be imported to picture recognition target.
<b>UPLOAD_FILEVOLUME_DATA</b>	38	File volume file
<b>IMPORT_DATA_TO_FACELIB</b>	39	Face data (face picture and picture additional information) to be imported to face picture library of device.
<b>UPLOAD_LEFTEYE_4K_CALIBFILE</b>	40	Camera calibration parameter file.
<b>UPLOAD_SECURITY_CFG_FILE</b>	41	Configuration file to be securely imported.
<b>UPLOAD_RIGHT_CONTROLLER_AUDIO</b>	42	Audio file of main controller.
<b>UPLOAD_MODBUS_CFG_FILE</b>	43	Configuration file of Modbus protocol.
<b>UPLOAD_NOTICE_VIDEO_DATA</b>	44	Bulletin video file.
<b>UPLOAD_RS485_PROTOCOL_DLL_FILE</b>	45	Dynamic library file of RS485 protocol.
<b>UPLOAD_PIC_BY_BUF</b>	46	Picture file for importing by picture cache.
<b>UPLOAD_CLIENT_CALIBFILE</b>	47	User calibration file (PTO format).

Enumeration Type	Macro Definition Value	Description
UPLOAD_HD_CAMERA_CORRECT_TABLE_3200W	48	HD camera correction sheet (CAL format).
UPLOAD_DOOR_CONTENT	49	Contact information of the door at the building unit.
UPLOAD_ASR_CONTROL_FILE	50	Speech recognition control file.
UPLOAD_APP_FILE	51	Application program file.
UPLOAD_AI_ALGORITHM_MODEL	52	Algorithm model in binary format.
UPLOAD_AI_BASE_PICTURE	55	Reference pictures in binary format for AI target comparison.
UPLOAD_OFFLINE_CAPTURE_INFO	56	User list of offline collection to be imported.
IMPORT_DATA_TO_HBDLIB	60	Import human body picture with linked information to library.
UPLOAD_SCENE_FILE	61	Scene file to be imported.
UPLOAD_RATIOSTITCHING_FILE	62	Ratio stitching file to be imported.
UPLOAD_LENS_PARAM_FILE	63	Lens parameters file to be imported.

## A.110 NET\_VCA\_POINT

### Structure About Point Coordinates Parameters

Member	Data Type	Description
fX	float	X-coordinate, it is a normalized value ranging from 0.000 to 1. The floating-point number is the percentage of the current image size and is accurate to three decimal places.
fY	float	Y-coordinate, it is a normalized value ranging from 0.000 to 1. The floating-point number is the percentage of the current image size and is accurate to three decimal places.

## A.111 NET\_VCA\_RECT

### Structure About Rectangle Region Coordinate Parameters

Member	Data Type	Description
fX	float	X-coordinate of frame's upper-left corner, it ranges from 0.000 to 1.
fY	float	Y-coordinate of frame' upper-left corner, it ranges from 0.000 to 1.
fWidth	float	Frame width, it ranges from 0.000 to 1.
fHeight	float	Frame height, it ranges from 0.000 to 1.

## Appendix B. Event Linkage Types

For event card linkages, if the linkage type is event, four major event linkage types are available: 0-device event linkage, 1-alarm input event linkage, 2-access control point (e.g., doors, elevators, etc.) event linkage, and 3-authentication unit (e.g., card reader, fingerprint module, etc.) event linkage. Each major event linkage type corresponds multiple minor types of event linkage, see details in the following content.

### Device Event Linkage

Minor Type	Value	Description
EVENT_ACS_HOST_ANTI_DISMANTLE	0	Access Controller Tampering Alarm
EVENT_ACS_OFFLINE_ECENT_NEARLY_FULL	1	No Memory Alarm
EVENT_ACS_NET_BROKEN	2	Network Disconnected
EVENT_ACS_NET_RESUME	3	Network Connected
EVENT_ACS_LOW_BATTERY	4	Low Battery Voltage
EVENT_ACS_BATTERY_RESUME	5	Battery Fully Charged
EVENT_ACS_AC_OFF	6	AC Power Off
EVENT_ACS_AC_RESUME	7	AC Power On
EVENT_ACS_SD_CARD_FULL	8	SD Card Full Alarm
EVENT_ACS_LINKAGE_CAPTURE_PIC	9	Capture Linkage Event Alarm
EVENT_ACS_IMAGE_QUALITY_LOW	10	Low Face Picture Quality
EVENT_ACS_FINGER_PRINT_QUALITY_LOW	11	Low Fingerprint Picture Quality
EVENT_ACS_BATTERY_ELECTRIC_LOW	12	Low Battery Voltage
EVENT_ACS_BATTERY_ELECTRIC_RESUME	13	Battery Fully Charged
EVENT_ACS_FIRE_IMPORT_SHORT_CIRCUIT	14	Fire Input Short Circuit Attempts Alarm

Minor Type	Value	Description
EVENT_ACS_FIRE_IMPORT_BROKEN_CIRCUIT	15	Fire Input Open Circuit Attempts Alarm
EVENT_ACS_FIRE_IMPORT_RESUME	16	Fire Input Alarm Restored
EVENT_ACS_MASTER_RS485_LOOPNODE_BROKEN	17	RS485 Loop of Main Access Controller Disconnected
EVENT_ACS_MASTER_RS485_LOOPNODE_RESUME	18	RS485 Loop of Main Access Controller Connected
EVENT_ACS_LOCAL_CONTROL_OFFLINE	19	Distributed Access Controller Offline
EVENT_ACS_LOCAL_CONTROL_RESUME	20	Distributed Access Controller Online
EVENT_ACS_LOCAL_DOWNSIDE_RS485_LOOPNODE_BROKEN	21	Downstream RS485 Loop of Distributed Access Control Disconnected
EVENT_ACS_LOCAL_DOWNSIDE_RS485_LOOPNODE_RESUME	22	Downstream RS485 Loop of Distributed Access Control Connected
EVENT_ACS_DISTRACT_CONTROLLER_ONLINE	23	Distributed Elevator Controller Online
EVENT_ACS_DISTRACT_CONTROLLER_OFFLINE	24	Distributed Elevator Controller Offline
EVENT_ACS_FIRE_BUTTON_TRIGGER	25	Fire Button Pressed
EVENT_ACS_FIRE_BUTTON_RESUME	26	Fire Button Released
EVENT_ACS_MAINTENANCE_BUTTON_TRIGGER	27	Maintenance Button Pressed
EVENT_ACS_MAINTENANCE_BUTTON_RESUME	28	Maintenance Button Released
EVENT_ACS_EMERGENCY_BUTTON_TRIGGER	29	Panic Button Pressed
EVENT_ACS_EMERGENCY_BUTTON_RESUME	30	Panic Button Released

Minor Type	Value	Description
EVENT_ACS_SUBMARINEBACK_COMM_BREAK	32	Communication with Anti-passing Back Server Failed
EVENT_ACS_SUBMARINEBACK_COMM_RESUME	33	Communication with Anti-passing Back Server Restored
EVENT_ACS_REMOTE_ACTUAL_GUARD	34	Remotely Armed
EVENT_ACS_REMOTE_ACTUAL_UNGUARD	35	Remotely Disarmed
EVENT_ACS_MOTOR_SENSOR_EXCEPTION	36	Motor or Sensor Exception
EVENT_ACS_CAN_BUS_EXCEPTION	37	CAN Bus Exception
EVENT_ACS_CAN_BUS_RESUME	38	CAN Bus Restored
EVENT_ACS_GATE_TEMPERATURE_OVERRUN	39	Too High Pedestal Temperature
EVENT_ACS_IR_EMITTER_EXCEPTION	40	Active Infrared Intrusion Detector Exception
EVENT_ACS_IR_EMITTER_RESUME	41	Active Infrared Intrusion Detector Restored
EVENT_ACS_LAMP_BOARD_COMM_EXCEPTION	42	Communication with Light Board Failed
EVENT_ACS_LAMP_BOARD_COMM_RESUME	43	Communication with Light Board Restored
EVENT_ACS_IR_ADAPTER_BOARD_COMM_EXCEPTION	44	Communication with IR Adaptor Failed
EVENT_ACS_IR_ADAPTER_BOARD_COMM_RESUME	45	Communication with IR Adaptor Restored
EVENT_ACS_CHANNEL_CONTROLLER_DESMANTLE_ALARM	46	Lane Controller Tampering Alarm

Minor Type	Value	Description
EVENT_ACS_CHANNEL_CONTROLLER_DESMANTLE_RESUME	47	Lane Controller Tampering Alarm Restored
EVENT_ACS_CHANNEL_CONTROLLER_FIRE_IMPORT_ALARM	48	Lane Controller Fire Input Alarm
EVENT_ACS_CHANNEL_CONTROLLER_FIRE_IMPORT_RESUME	49	Lane Controller Fire Input Alarm Restored
EVENT_ACS_STAY_EVENT	/	Staying Event
EVENT_ACS_LEGAL_EVENT_NEARLY_FULL	/	No Memory Alarm for Valid Offline Event Storage

## Alarm Input Event Linkage

Minor Type	Value	Description
EVENT_ACS_ALARMIN_SHORT_CIRCUIT	0	Zone Short Circuit Attempts Alarm
EVENT_ACS_ALARMIN_BROKEN_CIRCUIT	1	Zone Open Circuit Attempts Alarm
EVENT_ACS_ALARMIN_EXCEPTION	2	Zone Exception Alarm
EVENT_ACS_ALARMIN_RESUME	3	Zone Alarm Restored
EVENT_ACS_CASE_SENSOR_ALARM	4	Alarm Input Alarm
EVENT_ACS_CASE_SENSOR_RESUME	5	Alarm Input Alarm Restored

## Access Control Point Event Linkage

Minor Type	Value	Description
EVENT_ACS_LEADER_CARD_OPEN_BEGIN	0	Open Door with First Card Started
EVENT_ACS_LEADER_CARD_OPEN_END	1	Open Door with First Card Ended
EVENT_ACS_ALWAYS_OPEN_BEGIN	2	Remain Open Started
EVENT_ACS_ALWAYS_OPEN_END	3	Remain Open Ended
EVENT_ACS_ALWAYS_CLOSE_BEGIN	4	Remain Closed Started
EVENT_ACS_ALWAYS_CLOSE_END	5	Remain Closed Ended
EVENT_ACS_LOCK_OPEN	6	Door Unlocked
EVENT_ACS_LOCK_CLOSE	7	Door Locked
EVENT_ACS_DOOR_BUTTON_PRESS	8	Exit Button Pressed
EVENT_ACS_DOOR_BUTTON_RELEASE	9	Exit Button Released
EVENT_ACS_DOOR_OPEN_NORMAL	10	Door Open (Contact)
EVENT_ACS_DOOR_CLOSE_NORMAL	11	Door Closed (Contact)
EVENT_ACS_DOOR_OPEN_ANORMAL	12	Door Abnormally Open (Contact)
EVENT_ACS_DOOR_OPEN_TIMEOUT	13	Door Open Timed Out (Contact)
EVENT_ACS_REMOTE_OPEN_DOOR	14	Door Remotely Open
EVENT_ACS_REMOTE_CLOSE_DOOR	15	Door Remotely Closed

Minor Type	Value	Description
EVENT_ACS_REMOTE_ALWAYS_OPEN	16	Remain Open Remotely
EVENT_ACS_REMOTE_ALWAYS_CLOSE	17	Remain Closed Remotely
EVENT_ACS_NOT_BELONG_MULTI_GROUP	18	Card Not in Multiple Authentication Group
EVENT_ACS_INVALID_MULTI_VERIFY_PERIOD	19	Card Not in Multiple Authentication Duration
EVENT_ACS_MULTI_VERIFY_SUPER_RIGHT_FAIL	20	Multiple Authentication Mode: Super Password Authentication Failed
EVENT_ACS_MULTI_VERIFY_REMOTE_RIGHT_FAIL	21	Multiple Authentication Mode: Remote Authentication Failed
EVENT_ACS_MULTI_VERIFY_SUCCESS	22	Multiple Authentication Completed
EVENT_ACS_MULTI_VERIFY_NEED_REMOTE_OPEN	23	Multiple Authentication: Remotely Open Door
EVENT_ACS_MULTI_VERIFY_SUPERPASSWD_VERIFY_SUCCESS	24	Multiple Authentication: Super Password Authentication Completed
EVENT_ACS_MULTI_VERIFY_REPEAT_VERIFY_FAIL	25	Multiple Authentication: Repeated Authentication Failed
EVENT_ACS_MULTI_VERIFY_TIMEOUT	26	Multiple Authentication Timed Out
EVENT_ACS_REMOTE_CAPTURE_PIC	27	Remote Capture
EVENT_ACS_DOORBELL_RINGING	28	Doorbell Ring
EVENT_ACS_SECURITY_MODULE_DESMANTLE_ALARM	29	Secure Door Control Unit Tampering Alarm
EVENT_ACS_CALL_CENTER	30	Center Event
EVENT_ACS_FIRSTCARD_AUTHORIZE_BEGIN	31	First Card Authentication Started

Minor Type	Value	Description
EVENT_ACS_FIRSTCARD_AUTHORIZE_END	32	First Card Authentication End
EVENT_ACS_DOORLOCK_INPUT_SHORT_CIRCUIT	33	Lock Input Short Circuit Attempts Alarm
EVENT_ACS_DOORLOCK_INPUT_BROKEN_CIRCUIT	34	Lock Input Open Circuit Attempts Alarm
EVENT_ACS_DOORLOCK_INPUT_EXCEPTION	35	Lock Input Exception Alarm
EVENT_ACS_DOORCONTACT_INPUT_SHORT_CIRCUIT	36	Contact Input Short Circuit Attempts Alarm
EVENT_ACS_DOORCONTACT_INPUT_BROKEN_CIRCUIT	37	Contact Input Open Circuit Attempts Alarm
EVENT_ACS_DOORCONTACT_INPUT_EXCEPTION	38	Contact Input Exception Alarm
EVENT_ACS_OPENBUTTON_INPUT_SHORT_CIRCUIT	39	Exit Button Input Short Circuit Attempts Alarm
EVENT_ACS_OPENBUTTON_INPUT_BROKEN_CIRCUIT	40	Exit Button Input Open Circuit Attempts Alarm
EVENT_ACS_OPENBUTTON_INPUT_EXCEPTION	41	Exit Button Input Exception Alarm
EVENT_ACS_DOORLOCK_OPEN_EXCEPTION	42	Unlocking Exception
EVENT_ACS_DOORLOCK_OPEN_TIMEOUT	43	Unlocking Timed Out
EVENT_ACS_FIRSTCARD_OPEN_WITHOUT_AUTHORIZE	44	Unauthorized First Card Opening Failed
EVENT_ACS_CALL_LADDER_RELAY_BREAK	45	Call Elevator Relay Open
EVENT_ACS_CALL_LADDER_RELAY_CLOSE	46	Call Elevator Relay Closed
EVENT_ACS_AUTO_KEY_RELAY_BREAK	47	Auto Button Relay Open

Minor Type	Value	Description
EVENT_ACS_AUTO_KEY_RELAY_CLOSE	48	Auto Button Relay Closed
EVENT_ACS_KEY_CONTROL_RELAY_BREAK	49	Button Relay Open
EVENT_ACS_KEY_CONTROL_RELAY_CLOSE	50	Button Relay Closed
EVENT_ACS_REMOTE_VISITOR_CALL_LADDER	51	Visitor Calling Elevator
EVENT_ACS_REMOTE_HOUSEHOLD_CALL_LADDER	52	Resident Calling Elevator
EVENT_ACS_LEGAL_MESSAGE	52	Valid Message
EVENT_ACS_ILLEGAL_MESSAGE	53	Invalid Message
EVENT_ACS_TRAILING	54	Tailgating
EVENT_ACS_REVERSE_ACCESS	55	Reverse Passing
EVENT_ACS_FORCE_ACCESS	56	Force Collision
EVENT_ACS_CLIMBING_OVER_GATE	57	Climbing Over
EVENT_ACS_PASSING_TIMEOUT	58	Passing Timed Out
EVENT_ACS_INTRUSION_ALARM	59	Intrusion Alarm
EVENT_ACS_FREE_GATE_PASS_NOT_AUTH	60	Authentication Failed When Free Passing
EVENT_ACS_DROP_ARM_BLOCK	61	Barrier Obstructed
EVENT_ACS_DROP_ARM_BLOCK_RESUME	62	Barrier Restored
EVENT_ACS_REMOTE_CONTROL_CLOSE_DOOR	63	Door Closed via Keyfob

Minor Type	Value	Description
EVENT_ACS_REMOTE_CONTROL_OPEN_DOOR	64	Door Opened via Keyfob
EVENT_ACS_REMOTE_CONTROL_ALWAYS_OPEN_DOOR	65	Remain Open via Keyfob

## Authentication Unit Event Linkage

Minor Type	Value	Description
EVENT_ACS_STRESS_ALARM	0	Duress Alarm
EVENT_ACS_CARD_READER_DESMANTLE_ALARM	1	Card Reader Tampering Alarm
EVENT_ACS_LEGAL_CARD_PASS	2	Valid Card Authentication Completed
EVENT_ACS_CARD_AND_PSW_PASS	3	Card and Password Authentication Completed
EVENT_ACS_CARD_AND_PSW_FAIL	4	Card and Password Authentication Failed
EVENT_ACS_CARD_AND_PSW_TIMEOUT	5	Card and Password Authentication Timed Out
EVENT_ACS_CARD_MAX_AUTHENTICATE_FAIL	6	Card Authentication Attempts Reach Limit
EVENT_ACS_CARD_NO_RIGHT	7	No Permission for Card
EVENT_ACS_CARD_INVALID_PERIOD	8	Invalid Card Swiping Time Period
EVENT_ACS_CARD_OUT_OF_DATE	9	Expired Card
EVENT_ACS_INVALID_CARD	10	Card No. Not Exist
EVENT_ACS_ANTI_SNEAK_FAIL	11	Anti-passing Back Authentication Failed
EVENT_ACS_INTERLOCK_DOOR_NOT_CLOSE	12	Interlocking Door Not Closed

Minor Type	Value	Description
EVENT_ACS_FINGERPRINT_COMPARE_PASS	13	Fingerprint Matched
EVENT_ACS_FINGERPRINT_COMPARE_FAIL	14	Fingerprint Mismatched
EVENT_ACS_CARD_FINGERPRINT_VERIFY_PASS	15	Card and Fingerprint Authentication Completed
EVENT_ACS_CARD_FINGERPRINT_VERIFY_FAIL	16	Card and Fingerprint Authentication Failed
EVENT_ACS_CARD_FINGERPRINT_VERIFY_TIMEOUT	17	Card and Fingerprint Authentication Timed Out
EVENT_ACS_CARD_FINGERPRINT_PASSWD_VERIFY_PASS	18	Card, Fingerprint, and Password Authentication Completed
EVENT_ACS_CARD_FINGERPRINT_PASSWD_VERIFY_FAIL	19	Card and Fingerprint Authentication Failed
EVENT_ACS_CARD_FINGERPRINT_PASSWD_VERIFY_TIMEOUT	20	Card and Fingerprint Authentication Timed Out
EVENT_ACS_FINGERPRINT_PASSWD_VERIFY_PASS	21	Fingerprint and Password Authentication Completed
EVENT_ACS_FINGERPRINT_PASSWD_VERIFY_FAIL	22	Fingerprint and Password Authentication Failed
EVENT_ACS_FINGERPRINT_PASSWD_VERIFY_TIMEOUT	23	Fingerprint and Password Authentication Timed Out
EVENT_ACS_FINGERPRINT_INEXISTENCE	24	Fingerprint Not Exist
EVENT_ACS_EMPLOYEEENO_AND_FP_VERIFY_PASS	42	Employee ID and Fingerprint Authentication Completed
EVENT_ACS_EMPLOYEEENO_AND_FP_VERIFY_FAIL	43	Employee ID and Fingerprint Authentication Failed
EVENT_ACS_EMPLOYEEENO_AND_FP_VERIFY_TIMEOUT	44	Employee ID and Fingerprint Authentication Timed Out

Minor Type	Value	Description
EVENT_ACS_EMPLOYEEENO_AND_FP_AND_PW_VERIFY_PASS	45	Employee ID, Fingerprint, and Password Authentication Completed
EVENT_ACS_EMPLOYEEENO_AND_FP_AND_PW_VERIFY_FAIL	46	Employee ID, Fingerprint, and Password Authentication Failed
EVENT_ACS_EMPLOYEEENO_AND_FP_AND_PW_VERIFY_TIMEOUT	47	Employee ID, Fingerprint, and Password Authentication Timed Out
EVENT_ACS_EMPLOYEEENO_AND_PW_PASS	52	Employee ID and Password Authentication Completed
EVENT_ACS_EMPLOYEEENO_AND_PW_FAIL	52	Employee ID and Password Authentication Failed
EVENT_ACS_EMPLOYEEENO_AND_PW_TIMEOUT	53	Employee ID and Password Authentication Timed Out
EVENT_ACS_DOOR_OPEN_OR_DORMANT_FAIL	57	Authentication Failed When Door Remain Closed or Door in Sleeping Mode
EVENT_ACS_AUTH_PLAN_DORMANT_FAIL	58	Authentication Failed When Authentication Schedule in Sleeping Mode
EVENT_ACS_CARD_ENCRYPT_VERIFY_FAIL	59	Card Encryption Verification Failed
EVENT_ACS_SUBMARINEBACK_REPLY_FAIL	60	Anti-passing Back Server Response Failed
EVENT_ACS_PASSWORD_MISMATCH	61	Password Mismatched
EVENT_ACS_EMPLOYEE_NO_NOT_EXIST	62	Employee ID Not Exist
EVENT_ACS_COMBINED_VERIFY_PASS	63	Combined Authentication Completed
EVENT_ACS_COMBINED_VERIFY_TIMEOUT	64	Combined Authentication Timed Out

Minor Type	Value	Description
EVENT_ACS_VERIFY_MODE_MISMATCH	65	Authentication Type Mismatched
EVENT_ACS_PSW_ERROR_OVER_TIMES	67	Maximum Password Authentication Failure Attempts
EVENT_ACS_PSW_VERIFY_PASS	68	Password Authenticated
EVENT_ACS_PSW_VERIFY_FAIL	69	Password Authentication Failed
EVENT_ACS_ORCODE_VERIFY_PASS	70	QR Code Authenticated
EVENT_ACS_ORCODE_VERIFY_FAIL	71	QR Code Authentication Failed
EVENT_ACS_HOUSEHOLDER_AUTHORIZE_PASS	72	Resident Authorization Authenticated
EVENT_ACS_BLUETOOTH_VERIFY_PASS	73	Bluetooth Authenticated
EVENT_ACS_BLUETOOTH_VERIFY_FAIL	74	Bluetooth Authentication Failed
EVENT_ACS_INFORMAL_MIFARE_CARD_VERIFY_FAIL	/	Authentication Failed: Invalid Mifare Card
EVENT_ACS_CPU_CARD_ENCRYPT_VERIFY_FAIL	/	Verifying CPU Card Encryption Failed
EVENT_ACS_NFC_DISABLE_VERIFY_FAIL	/	Disabling NFC Verification Failed
EVENT_ACS_EM_CARD_RECOGNIZE_NOT_ENABLED	/	EM Card Recognition Disabled
EVENT_ACS_M1_CARD_RECOGNIZE_NOT_ENABLED	/	M1 Card Recognition Disabled
EVENT_ACS_CPU_CARD_RECOGNIZE_NOT_ENABLED	/	CPU Card Recognition Disabled
EVENT_ACS_ID_CARD_RECOGNIZE_NOT_ENABLED	/	ID Card Recognition Disabled
EVENT_ACS_CARD_SET_SECRET_KEY_FAIL	/	Importing Key to Card Failed

## Appendix C. HCNetSDK Log Types

The logs generated by the devices during the HCNetSDK integration are classified as five major types, i.e., alarm log (MAJOR\_ALARM-01), exception log (MAJOR\_EXCEPTION-0x2), operation log (MAJOR\_OPERATION-0x3), additional information log (MAJOR\_INFORMATION-0x4), and event log (MAJOR\_EVENT-0x5). Each major log type corresponds to multiple minor types, see details below.

### **MAJOR\_ALARM**

**Table B-1 Minor Types of Alarm Log**

Log Minor Type	Value	Description
MINOR_ALARM_IN	0x1	Alarm Input
MINOR_ALARM_OUT	0x2	Alarm output
MINOR_MOTDET_START	0x3	Motion detection alarm started
MINOR_MOTDET_STOP	0x4	Motion detection alarm ended
MINOR_HIDE_ALARM_START	0x5	Tampering alarm started
MINOR_HIDE_ALARM_STOP	0x6	Tampering alarm ended
MINOR_VCA_ALARM_START	0x7	VCA alarm started
MINOR_VCA_ALARM_STOP	0x8	VCA alarm ended
MINOR_ITS_ALARM_START	0x09	Traffic event alarm started
MINOR_ITS_ALARM_STOP	0x0a	Traffic event alarm ended
MINOR_NETALARM_START	0x0b	Network alarm started
MINOR_NETALARM_STOP	0x0c	Network alarm ended
MINOR_NETALARM_RESUME	0x0d	Network alarm recovery
MINOR_WIRELESS_ALARM_START	0x0e	Wireless alarm started
MINOR_WIRELESS_ALARM_STOP	0x0f	Wireless alarm ended
MINOR_PIR_ALARM_START	0x10	Human induction alarm started
MINOR_PIR_ALARM_STOP	0x11	Human induction alarm ended
MINOR_CALLHELP_ALARM_START	0x12	Emergency alarm started
MINOR_CALLHELP_ALARM_STOP	0x13	Emergency alarm ended
MINOR_DETECTFACE_ALARM_START	0x16	Face detection alarm started

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_DETECTFACE_ALARM_STOP	0x17	Face detection alarm ended
MINOR_VCA_SECNECHANGE_DETECTION	0x1a	Scene change detection alarm
MINOR_SMART_REGION_EXITING_BEGIN	0x1b	Region exiting detection started
MINOR_SMART_REGION_EXITING_END	0x1c	Region exiting detection ended
MINOR_SMART_LOITERING_BEGIN	0x1d	Loitering detection started
MINOR_SMART_LOITERING_END	0x1e	Loitering detection ended
MINOR_DREDGERDETECTION_ALARM	0x11a	Dredger detection alarm
MINOR_VCA_ALARM_LINE_DETECTION_BEGIN	0x20	Line crossing detection started
MINOR_VCA_ALARM_LINE_DETECTION_END	0x21	Line crossing detection ended
MINOR_VCA_ALARM_INTRUDE_BEGIN	0x22	Intrusion detection started
MINOR_VCA_ALARM_INTRUDE_END	0x23	Intrusion detection ended
MINOR_VCA_ALARM_AUDIOINPUT	0x24	Audio loss detection
MINOR_VCA_ALARM_AUDIOABNORMAL	0x25	Audio exception detection
MINOR_VCA_DEFOCUS_DETECTION_BEGIN	0x26	Defocus detection started
MINOR_VCA_DEFOCUS_DETECTION_END	0x27	Defocus detection ended
MINOR_VCA_FACE_ALARM_BEGIN	0x29	Face detection started
MINOR_SMART_REGION_ENTRANCE_BEGIN	0x2a	Region entrance detection started
MINOR_SMART_REGION_ENTRANCE_END	0x2b	Region entrance detection ended
MINOR_SMART_PEOPLE_GATHERING_BEGIN	0x2c	People gathering detection started

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_SMART_PEOPLE_GATHERING_END	0x2d	People gathering detection ended
MINOR_SMART_FAST_MOVING_BEGIN	0x2e	Fast moving detection started
MINOR_SMART_FAST_MOVING_END	0x2f	Fast moving detection ended
MINOR_VCA_FACE_ALARM_END	0x30	Face detection ended
MINOR_VCA_SCENE_CHANGE_ALARM_BEGIN	0x31	Scene change detection started
MINOR_VCA_SCENE_CHANGE_ALARM_END	0x32	Scene change detection ended
MINOR_VCA_ALARM_AUDIOINPUT_BEGIN	0x33	Audio loss detection started
MINOR_VCA_ALARM_AUDIOINPUT_END	0x34	Audio loss detection ended
MINOR_VCA_ALARM_AUDIOABNORMAL_BEGIN	0x35	Sudden change of sound intensity detection started
MINOR_VCA_ALARM_AUDIOABNORMAL_END	0x36	Sudden change of sound intensity detection ended
MINOR_VCA_ALARM_AUDIOSTEEPDROP	0x39	Sudden decrease of sound intensity detection
MINOR_SMART_PARKING_BEGIN	0x3c	Parking detection started
MINOR_SMART_PARKING_END	0x3d	Parking detection ended
MINOR_SMART_UNATTENDED_BAGGAGE_BEGIN	0x3e	Unattended baggage detection started
MINOR_SMART_UNATTENDED_BAGGAGE_END	0x3f	Unattended baggage detection ended
MINOR_SMART_OBJECT_REMOVAL_BEGIN	0x40	Object removal detection started
MINOR_SMART_OBJECT_REMOVAL_END	0x41	Object removal detection ended
MINOR_VCA_LEAVE_POSITION_START	0x42e	Absence detection started
MINOR_VCA_LEAVE_POSITION_STOP	0x42f	Absence detection ended

Log Minor Type	Value	Description
MINOR_VCA_PEOPLENUM_CHANGE_START	0x434	The people number change started
MINOR_VCA_PEOPLENUM_CHANGE_STOP	0x435	The people number change ended
MINOR_VCA_RUNNING_START	0x438	People running started
MINOR_VCA_RUNNING_STOP	0x439	People running ended
MINOR_VCA_VIOLENT_MOTION_START	0x43a	Violent motion started
MINOR_VCA_VIOLENT_MOTION_STOP	0x43b	Violent motion ended
MINOR_VCA_FAIL_DOWN_START	0x43c	People falling started
MINOR_VCA_FAIL_DOWN_STOP	0x43d	People falling ended
MINOR_VCA_RETENTION_START	0x43e	Overstay detection started
MINOR_VCA_RETENTION_STOP	0x43f	Overstay detection ended
MINOR_SMART_VEHICLE_ALARM_START	0x46	License plate detection started
MINOR_SMART_VEHICLE_ALARM_STOP	0x47	License plate detection ended
MINOR_THERMAL_FIREDETECTION	0x48	Thermal imaging fire point detection started
MINOR_THERMAL_FIREDETECTION_END	0x49	Thermal imaging fire point detection ended
MINOR_SMART_VANDALPROOF_BEGIN	0x50	Vandal-proof detection started
MINOR_SMART_VANDALPROOF_END	0x51	Vandal-proof detection ended
MINOR_FACESHAP_MATCH_ALARM_START	0x55	Face picture comparison alarm started
MINOR_FACESHAP_MATCH_ALARM_STOP	0x56	Face picture comparison alarm ended
MINOR_ALLOWLIST_FACESHAP_MATCH_ALARM_START	0x57	Face picture in allowlist comparison alarm started

Log Minor Type	Value	Description
MINOR_ALLOWLIST_FACESNAP_MATCH_ALARM_STOP	0x58	Face picture in allowlist comparison alarm ended
MINOR_THERMAL_SHIPSDETECTION	0x5a	Thermal imaging ship detection
MINOR_THERMAL_THERMOMETRY_EARLYWARNING_BEGIN	0x5b	Thermal imaging temperature measurement pre-alarm started
MINOR_THERMAL_THERMOMETRY_EARLYWARNING_END	0x5c	Thermal imaging temperature measurement pre-alarm ended
MINOR_THERMAL_THERMOMETRY_ALARM_BEGIN	0x5d	Thermal imaging temperature measurement alarm started
MINOR_THERMAL_THERMOMETRY_ALARM_END	0x5e	Thermal imaging temperature measurement alarm ended
MINOR_THERMAL_THERMOMETRY_DIFF_ALARM_BEGIN	0x5f	Thermal imaging temperature difference alarm started
MINOR_THERMAL_THERMOMETRY_DIFF_ALARM_END	0x60	Thermal imaging temperature difference alarm ended
MINOR_FACE_THERMOMETRY_ALARM	0x63	Body thermometry alarm
MINOR_SAFETY_HELMET_ALARM_START	0x72	Hard hat detection alarm stated
MINOR_SAFETY_HELMET_ALARM_STOP	0x73	Hard hat detection alarm ended
MINOR_HFPD_ALARM_START	0x74	Frequently appeared person detection alarm started
MINOR_HFPD_ALARM_STOP	0x75	Frequently appeared person detection alarm ended
MINOR_MIXED_TARGET_ALARM_START	0x76	Milti-target-type detection alarm started
MINOR_MIXED_TARGET_ALARM_STOP	0x77	Milti-target-type detection alarm ended
MINOR_VCA_GET_UP_ALARM_BEGIN	0x80	Getting up alarm started
MINOR_VCA_GET_UP_ALARM_END	0x81	Getting up alarm ended
MINOR_VCA_ADV_REACH_HEIGHT_ALARM_BEGIN	0x82	Climbing alarm started

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_VCA_ADV_REACH_HEIGHT_ALARM_END	0x83	Climbing alarm ended
MINOR_VCA_TOILET_TARRY_ALARM_BEGIN	0x84	Toilet overtime alarm started
MINOR_VCA_TOILET_TARRY_ALARM_END	0x85	Toilet overtime alarm ended
MINOR_HUMAN_RECOGNITION_ALARM_BEGIN	0x86	Target alarm started
MINOR_HUMAN_RECOGNITION_ALARM_END	0x87	Target alarm ended
MINOR_ACCESS_CONTROLLER_EVENT	0x100	Access controller event
MINOR_VIDEO_INTERCOM_EVENT	0x101	Video intercom event
MINOR_GJD_EVENT	0x102	GJD security control panel event
MINOR_LUMINITE_EVENT	0x103	LUMINITE security control panel event
MINOR_OPTEX_EVENT	0x104	OPTEX security control panel event
MINOR_CAMERA_DETECTOR_EVENT	0x105	Detector event
MINOR_SECURITY_CONTROL_PANEL_EVENT	0x106	Security control panel event
MINOR_LFPD_ALARM_START	0x124	Low frequency person alarm started
MINOR_LFPD_ALARM_STOP	0x125	Low frequency person alarm stopped
MINOR_DATA_PREALARM_ALARM	0x127	Network traffic pre-alarm
MINOR_VIBRATION_DETECTION_ALARM_BEGIN	0x132	Vibration detection alarm started
MINOR_VIBRATION_DETECTION_ALARM_END	0x133	Vibration detection alarm stopped
MINOR_ALARMIN_SHORT_CIRCUIT	0x400	Zone short circuited alarm
MINOR_ALARMIN_BROKEN_CIRCUIT	0x401	Zone disconnected alarm
MINOR_ALARMIN_EXCEPTION	0x402	Zone exception alarm
MINOR_ALARMIN_RESUME	0x403	Zone alarm recovery

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_HOST_DESMANTLE_ALARM	0x404	Device anti-tamper alarm
MINOR_HOST_DESMANTLE_RESUME	0x405	Device anti-tamper recovery
MINOR_CARD_READER_DESMANTLE_ALARM	0x406	Card reader anti-tamper alarm
MINOR_CARD_READER_DESMANTLE_RESUME	0x407	Card reader anti-tamper recovery
MINOR_CASE_SENSOR_ALARM	0x408	Event input alarm
MINOR_CASE_SENSOR_RESUME	0x409	Event input recovery
MINOR_STRESS_ALARM	0x40a	Duress alarm
MINOR_OFFLINE_ECENT_NEARLY_FULL	0x40b	No memory alarm
MINOR_CARD_MAX_AUTHENTICATE_FAIL	0x40c	Card reading failure alarm
MINOR_POS_START_ALARM	0x411	POS enabled
MINOR_POS_END_ALARM	0x412	POS disabled

## MAJOR\_EXCEPTION

**Table B-2 Minor Types of Exception Log**

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_RAID_ERROR	0x20	RAID exception
MINOR_VI_LOST	0x21	Video loss
MINOR_ILLEGAL_ACCESS	0x22	Illegal login
MINOR_HD_FULL	0x23	HDD full
MINOR_HD_ERROR	0x24	HDD error
MINOR_DCD_LOST	0x25	MODEM offline (reserved)
MINOR_IP_CONFLICT	0x26	IP address conflicted
MINOR_NET_BROKEN	0x27	Network disconnected
MINOR_REC_ERROR	0x28	Recording error
MINOR_IPC_NO_LINK	0x29	IPC connection exception

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_VI_EXCEPTION	0x2a	Video input exception (only for analog channel)
MINOR_IPC_IP_CONFLICT	0x2b	IP address conflicted of IPC
MINOR_SENSE_EXCEPTION	0x2c	Sense exception
MINOR_PIC_REC_ERROR	0x2d	Capture error. Failed to get pictures.
MINOR_VI_MISMATCH	0x2e	Video format mismatches
MINOR_RESOLUTION_MISMATCH	0x2f	Encoding resolution does not match with the front-end resolution
MINOR_RS485_DEVICE_ABNORMAL	0x3a	RS485 connection status exception
MINOR_RS485_DEVICE_REVERT	0x3b	RS485 connection status exception recovery
MINOR_SCREEN_SUBSYSTEM_ABNORMALREBOOT	0x3c	Sub-board abnormal startup
MINOR_SCREEN_SUBSYSTEM_ABNORMALINSERT	0x3d	Sub-board inserted
MINOR_SCREEN_SUBSYSTEM_ABNORMALPULLOUT	0x3e	Sub-board pulled out
MINOR_SCREEN_ABNARMALTEMPERATURE	0x3f	Temperature exception
MINOR_RECORD_OVERFLOW	0x41	Buffer overflow
MINOR_DSP_ABNORMAL	0x42	DSP exception
MINOR_ANR_RECORD_FAIED	0x43	ANR recording failed
MINOR_SPARE_WORK_DEVICE_EXCEPT	0x44	Hot spare device working exception
MINOR_START_IPC_MAS_FAILED	0x45	Failed to enable IPC MAS
MINOR_IPCM_CRASH	0x46	IPCM abnormal rebooting
MINOR_POE_POWER_EXCEPTION	0x47	POE power supply exception
MINOR_UPLOAD_DATA_CS_EXCEPTION	0x48	Failed to upload data to cloud storage
MINOR_DIAL_EXCEPTION	0x49	Dial-up exception
MINOR_DEV_EXCEPTION_OFFLINE	0x50	Device abnormal offline

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_UPGRADEFAIL	0x51	Remote upgrading failed.
MINOR_AI_LOST	0x52	Audio loss
MINOR_SYNC_IPC_PASSWD	0x53	IPC password synchronization exception
MINOR_EZVIZ_OFFLINE	0x54	Ezviz offline exception
MINOR_ACCESSORIES_PLATE	0x57	Accessory board exception
MINOR_CAMERA_ANGLE_ANOMALY	0x60	Camera view angle exception
MINOR_FACESHAP_RESOLUTION_OVERFLOW	0x63	Overlimit face capture stream resolution
MINOR_SMD_RESOLUTION_OVERFLOW	0x64	Overlimit SMD stream resolution
MINOR_AUDIO_LOSS_EXCEPTION	0x65	Audio loss
MINOR_SAFETY_HELMET_EXCEPTION	0x66	Hard hat detection exception
MINOR_VCA_PIC_LENGTH_OVERFLOW	0x67	The VCA picture size is too large
MINOR_FACE_MODEL_EXCEPTION	0x68	Face picture library model synchronization error
MINOR_CLUSTER_DEVICE_OFFLINE	0x70	The device in cluster is offline
MINOR_CLUSTER_CONFIG_FAILED	0x71	Configuring the devices in cluster failed.
MINOR_CLUSTER_DISASTER_TOLERANCE_EXCEPT	0x72	Cluster disaster recovery exception: cluster CM election failed, no enough cluster storage period, no enough cluster bandwidth, no enough channel resource, no enough device.
MINOR_CLUSTER_STORFULL_EXCEPTION	0x73	The cluster HDD is full.
MINOR_CLUSTER_VERSION_EXCEPTION	0x74	Cluster version exception
MINOR_CLUSTER_OFFLINENODE_EXCEPTION	0x75	The offline devices in cluster exceed the limit.
MINOR_CLUSTER_RECORDCYCLE_EXCEPTION	0x76	Cluster storage period is not enough.

Log Minor Type	Value	Description
MINOR_CLUSTER_IPCTRANSFER_EXCEPTION	0x77	Cluster network camera migration failed.
MINOR_CLUSTER_IPCONFLICT_EXCEPTION	0x78	Cluster IP conflict.
MINOR_EVENT_UPLOAD_EXCEPTION	0x7c	Uploading event failed/Uploaded event lost
MINOR_DEV_POWER_ON	0x400	Device power on
MINOR_DEV_POWER_OFF	0x401	Device power off
MINOR_WATCH_DOG_RESET	0x402	Watch dog resumed
MINOR_LOW_BATTERY	0x403	Low battery
MINOR_BATTERY_RESUME	0x404	Battery voltage recovery
MINOR_AC_OFF	0x405	AC power interrupt
MINOR_AC_RESUME	0x406	AC power recovery
MINOR_NET_RESUME	0x407	Network recovery
MINOR_FLASH_ABNORMAL	0x408	FLASH reading/writing exception
MINOR_CARD_READER_OFFLINE	0x409	Card reader offline
MINOR_CARD_READER_RESUME	0x40a	Card reader offline recovery
MINOR_DSP_START_FAILED	0x43a	Starting up DSP failed.
MINOR_SMART_REGULATION_NOT_ALLOWED	0x43b	Intelligent rule is not supported.
MINOR_AUXILIARY_BOARD_OFFLINE	0x43c	Auxiliary board disconnected
MINOR_AUXILIARY_BOARD_RESUME	0x43d	Auxiliary board connected
MINOR_IDCARD_SECURITY_MOUDLE_EXCEPTION	0x43e	ID card module exception
MINOR_IDCARD_SECURITY_MOUDLE_RESUME	0x43f	ID card module restored
MINOR_FP_PERIPHERAL_EXCEPTION	0x440	Fingerprint recorder exception
MINOR_FP_PERIPHERAL_RESUME	0x441	Fingerprint recorder restored
MINOR_SUBSYSTEM_IP_CONFLICT	0x4000	IP conflicted of sub-board
MINOR_SUBSYSTEM_NET_BROKEN	0x4001	Sub-board offline

Log Minor Type	Value	Description
MINOR_FAN_ABNORMAL	0x4002	Fan exception
MINOR_BACKPANEL_TEMPERATURE_ABNORMAL	0x4003	Back board temperature exception
MINOR_SDCARD_ABNORMAL	0x4004	SD card defective
MINOR_SDCARD_DAMAGE	0x4005	SD card damaged
MINOR_OVERTVOLTAGE	0x4019	High supply voltage
MINOR_UNDERVOLTAGE	0x401a	Low supply voltage
MINOR_EZVIZ_UPGRADE_EXCEPTION	0x401e	Guarding Vision upgrade exception
MINOR_HIGH_HD_TEMPERATURE	0x80	HDD high temperature
MINOR_LOW_HD_TEMPERATURE	0x81	HDD low temperature
MINOR_HD_IMPACT	0x82	HDD impact
MINOR_HD_BAD_BLOCK	0x83	HDD bad sector
MINOR_SEVERE_HD_FAILURE	0x84	HDD severe fault

## MAJOR\_OPERATION

Table B-3 Minor Types of Operation Log

Log Minor Type	Value	Description
MINOR_START_DVR	0x41	Power on
MINOR_STOP_DVR	0x42	Shutdown
MINOR_STOP_ABNORMAL	0x43	Abnormal shutdown
MINOR_REBOOT_DVR	0x44	Reboot device (local)
MINOR_LOCAL_LOGIN	0x50	Logged in (local)
MINOR_LOCAL_LOGOUT	0x51	Logged out (Local)
MINOR_LOCAL_CFG_PARM	0x52	Local configuration
MINOR_LOCAL_PLAYBYFILE	0x53	Playback or download by file (local)
MINOR_LOCAL_PLAYBYTIME	0x54	Playback or download by time (local)
MINOR_LOCAL_START_REC	0x55	Start recording (local)
MINOR_LOCAL_STOP_REC	0x56	Stop recording (local)
MINOR_LOCAL_PTZCTRL	0x57	PTZ control (local)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_PREVIEW	0x58	Live view (local,reserved)
MINOR_LOCAL MODIFY_TIME	0x59	Edit time (local,reserved)
MINOR_LOCAL_UPGRADE	0x5a	Local upgrade
MINOR_LOCAL_RECFILE_OUTPUT	0x5b	Backup video files (local)
MINOR_LOCAL_FORMAT_HDD	0x5c	Initialize HDD (local)
MINOR_LOCAL_CFGFILE_OUTPUT	0x5d	Export local configuration files
MINOR_LOCAL_CFGFILE_INPUT	0x5e	Import local configuration files
MINOR_LOCAL_COPYFILE	0x5f	Backup files (local)
MINOR_LOCAL_LOCKFILE	0x60	Lock video files (local)
MINOR_LOCAL_UNLOCKFILE	0x61	Unlock video files (local)
MINOR_LOCAL_DVR_ALARM	0x62	Clear manually and trigger alarm (local)
MINOR_IPC_ADD	0x63	Add IPC (local)
MINOR_IPC_DEL	0x64	Delete IPC (local)
MINOR_IPC_SET	0x65	Set IPC (local)
MINOR_LOCAL_START_BACKUP	0x66	Start backup (local)
MINOR_LOCAL_STOP_BACKUP	0x67	Stop backup (local)
MINOR_LOCAL_COPYFILE_START_TIME	0x68	Start time of local backup
MINOR_LOCAL_COPYFILE_END_TIME	0x69	End time of local backup
MINOR_LOCAL_ADD_NAS	0x6a	Add NetHDD (local)
MINOR_LOCAL_DEL_NAS	0x6b	Delete NAS (local)
MINOR_LOCAL_SET_NAS	0x6c	Set NAS (local)
MINOR_REMOTE_LOGIN	0x70	Login (remote)
MINOR_REMOTE_LOGOUT	0x71	Logout (local)
MINOR_REMOTE_START_REC	0x72	Start recording (remote)
MINOR_REMOTE_STOP_REC	0x73	Stop recording (remote)
MINOR_START_TRANS_CHAN	0x74	Start transparent transmission
MINOR_STOP_TRANS_CHAN	0x75	Stop transparent transmission

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_REMOTE_GET_PARM	0x76	Get parameters (remote)
MINOR_REMOTE_CFG_PARM	0x77	Remote configuration
MINOR_REMOTE_GET_STATUS	0x78	Get status (remote)
MINOR_REMOTE_ARM	0x79	Arm (remote)
MINOR_REMOTE_DISARM	0x7a	Disarm (remote)
MINOR_REMOTE_REBOOT	0x7b	Reboot (remote)
MINOR_START_VT	0x7c	Start two-way audio
MINOR_STOP_VT	0x7d	Stop two-way audio
MINOR_REMOTE_UPGRADE	0x7e	Remote upgrade
MINOR_REMOTE_PLAYBYFILE	0x7f	Playback by file (remote)
MINOR_REMOTE_PLAYBYTIME	0x80	Playback by time (remote)
MINOR_REMOTE_PTZCTRL	0x81	PTZ control (remote)
MINOR_REMOTE_FORMAT_HDD	0x82	Format HDD (remote)
MINOR_REMOTE_STOP	0x83	Shutdown (remote)
MINOR_REMOTE_LOCKFILE	0x84	Lock files (remote)
MINOR_REMOTE_UNLOCKFILE	0x85	Unlock files (remote)
MINOR_REMOTE_CFGFILE_OUTPUT	0x86	Export configuration files (remote)
MINOR_REMOTE_CFGFILE_INPUT	0x87	Import configuration files (remote)
MINOR_REMOTE_RECFILE_OUTPUT	0x88	Export video files (remote)
MINOR_REMOTE_DVR_ALARM	0x89	Clear manually and trigger alarm (remote)
MINOR_REMOTE_IPC_ADD	0x8a	Add IPC (remote)
MINOR_REMOTE_IPC_DEL	0x8b	Delete IPC (remote)
MINOR_REMOTE_IPC_SET	0x8c	Set IPC (remote)
MINOR_REMOTE_VCA_LIB	0x8d	Reboot intelligent library
MINOR_REMOTE_ADD_NAS	0x8e	Add NAS (remote)
MINOR_REMOTE_DEL_NAS	0x8f	Delete NAS (remote)
MINOR_REMOTE_SET_NAS	0x90	Set NAS (remote)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_START_REC_CDRW	0x91	Start burning (local)
MINOR_LOCAL_STOP_REC_CDRW	0x92	Stop burning (local)
MINOR_REMOTE_START_REC_CDRW	0x93	Start burning (remote)
MINOR_REMOTE_STOP_REC_CDRW	0x94	Stop burning (remote)
MINOR_LOCAL_PIC_OUTPUT	0x95	Back up pictures (local)
MINOR_REMOTE_PIC_OUTPUT	0x96	Back up pictures (remote)
MINOR_LOCAL_INQUEST_RESUME	0x97	Resume inquest event (local)
MINOR_REMOTE_INQUEST_RESUME	0x98	Resume inquest event (remote)
MINOR_LOCAL_ADD_FILE	0x99	Import files (local)
MINOR_REMOTE_DELETE_HDISK	0x9a	Delete exception or nonexistent HDD
MINOR_REMOTE_LOAD_HDISK	0x9b	Load HDD (remote)
MINOR_REMOTE_UNLOAD_HDISK	0x9c	Unload HDD (remote)
MINOR_LOCAL_OPERATE_LOCK	0x9d	Lock (local)
MINOR_LOCAL_OPERATE_UNLOCK	0x9e	Unlock (local)
MINOR_LOCAL_DEL_FILE	0x9f	Delete inquest files (local)
MINOR_REMOTE_BYPASS	0xd0	Bypass (remote)
MINOR_REMOTE_UNBYPASS	0xd1	Bypass recovery (remote)
MINOR_REMOTE_SET_ALARMIN_CFG	0xd2	Set alarm input parameters (remote)
MINOR_REMOTE_GET_ALARMIN_CFG	0xd3	Get alarm input parameters (remote)
MINOR_REMOTE_SET_ALARMOUT_CFG	0xd4	Set alarm output parameters (remote)
MINOR_REMOTE_GET_ALARMOUT_CFG	0xd5	Get alarm output parameters (remote)
MINOR_REMOTE_ALARMOUT_OPEN_MAN	0xd6	Enable alarm output manually (remote)
MINOR_REMOTE_ALARMOUT_CLOSE_MAN	0xd7	Disable alarm output manually (remote)
MINOR_REMOTE_ALARM_ENABLE_CFG	0xd8	Enable/Disable RS-485 serial port of security control panel (remote)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_DBDATA_OUTPUT	0xd9	Export database records
MINOR_DBDATA_INPUT	0xda	Import database records
MINOR_MU_SWITCH	0xdb	Cascading switch
MINOR_MU_PTZ	0xdc	Cascading PTZ control
MINOR_REMOTE_INQUEST_DEL_FILE	0xde	Delete file (remote)
MINOR_LOCAL_CONF_REB_RAID	0x101	Configure auto-rebuild (local)
MINOR_LOCAL_CONF_SPARE	0x102	Configure hot spare (local)
MINOR_LOCAL_ADD_RAID	0x103	Create array (local)
MINOR_LOCAL_DEL_RAID	0x104	Delete array (local)
MINOR_LOCAL_MIG_RAID	0x105	Migrate array (local)
MINOR_LOCAL_REB_RAID	0x106	Rebuild array manually (local)
MINOR_LOCAL_QUICK_CONF_RAID	0x107	One-touch configuration (local)
MINOR_LOCAL_ADD_VD	0x108	Create virtual disk (local)
MINOR_LOCAL_DEL_VD	0x109	Delete virtual disk (local)
MINOR_LOCAL_RP_VD	0x10a	Repair virtual disk (local)
MINOR_LOCAL_FORMAT_EXPANDVD	0x10b	Expand virtual disk (local)
MINOR_LOCAL_RAID_UPGRADE	0x10c	Upgrade RAID (local)
MINOR_LOCAL_STOP_RAID	0x10d	Pause RAID (local, unplug safely)
MINOR_REMOTE_CONF_REB_RAID	0x111	Configure auto-rebuild (remote)
MINOR_REMOTE_CONF_SPARE	0x112	Configure hot spare (remote)
MINOR_REMOTE_ADD_RAID	0x113	Create array (remote)
MINOR_REMOTE_DEL_RAID	0x114	Delete array (remote)
MINOR_REMOTE_MIG_RAID	0x115	Migrate array (remote)
MINOR_REMOTE_REB_RAID	0x116	Rebuild array manually (remote)
MINOR_REMOTE_QUICK_CONF_RAID	0x117	One-touch configuration (remote)
MINOR_REMOTE_ADD_VD	0x118	Create virtual disk (remote)
MINOR_REMOTE_DEL_VD	0x119	Delete virtual disk (remote)
MINOR_REMOTE_RP_VD	0x11a	Repair virtual disk (remote)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_REMOTE_FORMAT_EXPANDVD	0x11b	Expand virtual disk (remote)
MINOR_REMOTE_RAID_UPGRADE	0x11c	Upgrade RAID (remote)
MINOR_REMOTE_STOP_RAID	0x11d	Pause RAID (remote, unplug safely)
MINOR_LOCAL_START_PIC_REC	0x121	Start capture (local)
MINOR_LOCAL_STOP_PIC_REC	0x122	Stop capture (local)
MINOR_LOCAL_SET_SNMP	0x125	Set SNMP (local)
MINOR_LOCAL_TAG_OPT	0x126	Tag operation (local)
MINOR_REMOTE_START_PIC_REC	0x131	Start capture (remote)
MINOR_REMOTE_STOP_PIC_REC	0x132	Stop capture (remote)
MINOR_REMOTE_SET_SNMP	0x135	Set SNMP (remote)
MINOR_REMOTE_TAG_OPT	0x136	Tag operation (remote)
MINOR_SCHEDULE_ANGLECALIBRATION	0x139	Scheduled angle calibration
MINOR_LOCAL_VOUT_SWITCH	0x140	Switch output (local)
MINOR_STREAM_CABAC	0x141	Encoding performance configuration
MINOR_LOCAL_SPARE_OPT	0x142	N+1 hot spare operation (local)
MINOR_REMOTE_SPARE_OPT	0x143	N+1 hot spare operation (remote)
MINOR_LOCAL_IPCCFGFILE_OUTPUT	0x144	Export IPC configuration file (local)
MINOR_LOCAL_IPCCFGFILE_INPUT	0x145	Import IPC configuration file (local)
MINOR_LOCAL_IPC_UPGRADE	0x146	Upgrade IPC (local)
MINOR_REMOTE_IPCCFGFILE_OUTPUT	0x147	Export IPC configuration file (remote)
MINOR_REMOTE_IPCCFGFILE_INPUT	0x148	Import IPC configuration file (remote)
MINOR_REMOTE_IPC_UPGRADE	0x149	Upgrade IPC (remote)
MINOR_LOCAL_UNLOAD_HDISK	0x150	Uninstall HDD (local)
MINOR_LOCAL_AUDIO_MIX	0x151	Set audio mix parameters (local)
MINOR_REMOTE_AUDIO_MIX	0x152	Set audio mix parameters (remote)
MINOR_LOCAL_TRIAL_PAUSE	0x153	Pause inquest (local)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_TRIAL_RESUME	0x154	Resume inquest (local)
MINOR_REMOTE_TRIAL_PAUSE	0x155	Pause inquest (remote)
MINOR_REMOTE_TRIAL_RESUME	0x156	Resume inquest (remote)
MINOR_REMOTE MODIFY_ VERIFICATION_CODE	0x157	Change the verification code of the system
MINOR_SET_MULTI_MASTER	0x201	Set main screen of multi-screen controller
MINOR_SET_MULTI_SLAVE	0x202	Set sub-screen of multi-screen controller
MINOR_CANCEL_MULTI_MASTER	0x203	Cancel main screen of multi-screen controller
MINOR_CANCEL_MULTI_SLAVE	0x204	Cancel sub-screen of multi-screen controller
MINOR_SCREEN_SET_INPUT	0x251	Edit input source
MINOR_SCREEN_SET_OUTPUT	0x252	Edit output channel
MINOR_SCREEN_SET_OSD	0x253	Edit virtual LED
MINOR_SCREEN_SET_LOGO	0x254	Edit LOGO
MINOR_SCREEN_SET_LAYOUT	0x255	Set scene
MINOR_SCREEN_PICTUREPREVIEW	0x256	Display operation
MINOR_SCREEN_GET_OSD	0x257	Get virtual LED
MINOR_SCREEN_GET_LAYOUT	0x258	Get scene
MINOR_SCREEN_LAYOUT_CTRL	0x259	Scene control
MINOR_GET_ALL_VALID_WND	0x260	Get all the valid windows
MINOR_GET_SIGNAL_WND	0x261	Get single window information
MINOR_REMOTE_CLUSTER_MODE_CONFIG	0x261c	Remote operation: cluster mode configuration
MINOR_LOCAL_CLUSTER_MODE_CONFIG	0x261d	Local operation: cluster mode configuration
MINOR_REMOTE_CLUSTER_NETWORK_CONFIG	0x261e	Remote operation: NVR in cluster configuration

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_CLUSTER_NETWORK_CONFIG	0x261f	Local operation: NVR in cluster configuration
MINOR_REMOTE_CLUSTER_ADD_DEVICE	0x2620	Remote operation: Add device to cluster
MINOR_WINDOW_CTRL	0x262	Window control
MINOR_LOCAL_CLUSTER_ADD_DEVICE	0x2621	Local operation: Add device to cluster
MINOR_REMOTE_CLUSTER_DEL_DEVICE	0x2622	Remote operation: Delete device from cluster
MINOR_LOCAL_CLUSTER_DEL_DEVICE	0x2623	Local operation: Delete device from cluster
MINOR_REMOTE_HFPD_CFG	0x2624	Remote operation: frequently appeared person detection configuration
MINOR_REMOTE_FACE_CONTRAST_TASK	0x2625	Remote operation: face picture comparison task configuration
MINOR_REMOTE_LFPD_CFG	0x2626	Remote configuration of low frequency person detection
MINOR_REMOTE_IOTCFGFILE_INPUT	0x2627	Remote operation: import IoT configuration file
MINOR_REMOTE_IOTCFGFILE_OUTPUT	0x2628	Remote operation: export IoT configuration file
MINOR_LOCAL_IOT_ADD	0x2629	Local operation: add IoT channel
MINOR_REMOTE_IOT_ADD	0x262a	Remote operation: add IoT channel
MINOR_LOCAL_IOT_DEL	0x262b	Local operation: delete IoT channel
MINOR_REMOTE_IOT_DEL	0x262c	Remote operation: delete IoT channel
MINOR_LOCAL_IOT_SET	0x262d	Local operation: configure IoT channel
MINOR_REMOTE_IOT_SET	0x262e	Remote operation: configure IoT channel
MINOR_LOCAL_IOTCFGFILE_INPUT	0x262f	Local operation: import IoT configuration file
MINOR_LOCAL_IOTCFGFILE_OUTPUT	0x2630	Local operation: export IoT configuration file

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_GET_LAYOUT_LIST	0x263	Get scene list
MINOR_LAYOUT_CTRL	0x264	Scene control
MINOR_SET_LAYOUT	0x265	Set single scene
MINOR_GET_SIGNAL_LIST	0x266	Get input signal source list
MINOR_GET_PLAN_LIST	0x267	Get plan list
MINOR_SET_PLAN	0x268	Edit plan
MINOR_CTRL_PLAN	0x269	Control plan
MINOR_CTRL_SCREEN	0x270	Screen control
MINOR_ADD_NETSIG	0x271	Add signal source
MINOR_SET_NETSIG	0x272	Edit signal source
MINOR_SET_DECBDCFG	0x273	Set decoding board parameters
MINOR_GET_DECBDCFG	0x274	Get decoding board parameters
MINOR_GET_DEVICE_STATUS	0x275	Get device information
MINOR_UPLOAD_PICTURE	0x276	Upload background
MINOR_SET_USERPWD	0x277	Set password
MINOR_ADD_LAYOUT	0x278	Add scene
MINOR_DEL_LAYOUT	0x279	Delete scene
MINOR_DEL_NETSIG	0x280	Delete signal source
MINOR_ADD_PLAN	0x281	Add plan
MINOR_DEL_PLAN	0x282	Delete plan
MINOR_GET_EXTERNAL_MATRIX_CFG	0x283	Get external matrix settings
MINOR_SET_EXTERNAL_MATRIX_CFG	0x284	Set external matrix
MINOR_GET_USER_CFG	0x285	Get user settings
MINOR_SET_USER_CFG	0x286	Set user
MINOR_GET_DISPLAY_PANEL_LINK_CFG	0x287	Get video wall connection settings
MINOR_SET_DISPLAY_PANEL_LINK_CFG	0x288	Set video wall connection
MINOR_GET_WALLSCENE_PARAM	0x289	Get video wall scene

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_SET_WALLSCENE_PARAM	0x28a	Set video wall scene
MINOR_GET_CURRENT_WALLSCENE	0x28b	Get current scene
MINOR_SWITCH_WALLSCENE	0x28c	Scene switch
MINOR_LOCAL_LOAD_HDISK	0x300	Load HDD (local)
MINOR_LOCAL_DELETE_HDISK	0x301	Delete exception or nonexistence HDD (local)
MINOR_REMOTE_CFG_POE_WORK_MODE	0x361	Remotely set PoE working mode
MINOR_LOCAL_CFG_POE_WORK_MODE	0x362	Locally set PoE working mode
MINOR_REMOTE_CFG_FACE_CONTRAST	0x363	Remotely set face comparison
MINOR_LOCAL_CFG_FACE_CONTRAST	0x364	Locally set face comparison
MINOR_REMOTE_CFG_ALLOWLIST_FACE_CONTRAST	0x365	Remotely set face comparison in allowlist
MINOR_LOCAL_CHECK_TIME	0x367	Manual time synchronization (local)
MINOR_LOCAL_CFG_ALLOWLIST_FACE_CONTRAST	0x366	Locally set face comparison in allowlist
MINOR_REMOTE_CFG_WIRELESS_DIALPARAM	0x36c	Configure wireless dial-up parameters remotely
MINOR_LOCAL_CFG_WIRELESS_DIALPARAM	0x36d	Configure wireless dial-up parameters locally
MINOR_REMOTE_CFG_WIRELESS_SMSPARAM	0x36e	Configure wireless message parameters remotely
MINOR_LOCAL_CFG_WIRELESS_SMSPARAM	0x36f	Configure wireless message parameters locally
MINOR_REMOTE_CFG_WIRELESS_SMSSEIFHELP	0x370	Configure SMS self-service parameters remotely
MINOR_LOCAL_CFG_WIRELESS_SMSSEIFHELP	0x371	Configure SMS self-service parameters locally
MINOR_REMOTE_CFG_WIRELESS_NETFLOWPARAM	0x372	Configure wireless traffic parameters remotely

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_CFG_WIRELESS_NETFLOWPARAM	0x373	Configure wireless traffic parameters locally
MINOR_REMOTE_OPEN_DOOR	0x400	Open door (remote)
MINOR_REMOTE_CLOSE_DOOR	0x401	Close door (remote)
MINOR_REMOTE_ALWAYS_OPEN	0x402	Remain open (remote)
MINOR_REMOTE_ALWAYS_CLOSE	0x403	Remain closed (remote)
MINOR_REMOTE_CHECK_TIME	0x404	Manual time synchronization (remote)
MINOR_NTP_CHECK_TIME	0x405	NTP auto time synchronization
MINOR_REMOTE_CLEAR_CARD	0x406	Clear card No. (remote)
MINOR_REMOTE_RESTORE_CFG	0x407	Resume default parameters (remote)
MINOR_ALARMIN_ARM	0x408	Zone arming
MINOR_ALARMIN_DISARM	0x409	Zone disarming
MINOR_LOCAL_RESTORE_CFG	0x40a	Resume default parameters (local)
MINOR_OFFLINE_DATA_OUTPUT	0x423	Exported offline collection data
MINOR_CREATE_SSH_LINK	0x42d	Connected with SSH
MINOR_CLOSE_SSH_LINK	0x42e	Disconnected with SSH
MINOR_SET_TRIGGERMODE_CFG	0x1001	Set trigger mode parameters
MINOR_GET_TRIGGERMODE_CFG	0x1002	Get trigger mode parameters
MINOR_SET_IOOUT_CFG	0x1003	Set IO output parameters
MINOR_GET_IOOUT_CFG	0x1004	Get IO output parameters
MINOR_GET_TRIGGERMODE_DEFAULT	0x1005	Get recommended parameters of trigger mode
MINOR_GET_ITCSTATUS	0x1006	Get status detection parameters
MINOR_SET_STATUS_DETECT_CFG	0x1007	Set status detection parameters
MINOR_GET_STATUS_DETECT_CFG	0x1008	Get status detection parameters
MINOR_GET_VIDEO_TRIGGERMODE_CFG	0x1009	Get parameters of video e-police mode
MINOR_SET_VIDEO_TRIGGERMODE_CFG	0x100a	Set parameters of video e-police mode

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_WEB_AUTHENTICATION	0x111d	Web authentication method configuration
MINOR_HTTPS_ENABLED	0x111f	HTTPS switch configuration
MINOR_SET_NETWORK_CFG	0x112b	Set network parameters
MINOR_GET_NETWORK_CFG	0x112c	Get network parameters
MINOR_LOCAL_ADD_CAR_INFO	0x2001	Add vehicle information (local)
MINOR_LOCAL_MOD_CAR_INFO	0x2002	Edit vehicle information (local)
MINOR_LOCAL_DEL_CAR_INFO	0x2003	Delete vehicle information (local)
MINOR_LOCAL_FIND_CAR_INFO	0x2004	Search vehicle information (local)
MINOR_LOCAL_ADD_MONITOR_INFO	0x2005	Add arming information (local)
MINOR_LOCAL_MOD_MONITOR_INFO	0x2006	Edit arming information (local)
MINOR_LOCAL_DEL_MONITOR_INFO	0x2007	Delete arming information (local)
MINOR_LOCAL_FIND_MONITOR_INFO	0x2008	Search arming information (local)
MINOR_LOCAL_FIND_NORMAL_PASS_INFO	0x2009	Search normal passing information (local)
MINOR_LOCAL_FIND_ABNORMAL_PASS_INFO	0x200a	Search abnormal passing information (local)
MINOR_LOCAL_FIND_PEDESTRIAN_PASS_INFO	0x200b	Search normal passing information (local)
MINOR_LOCAL_PIC_PREVIEW	0x200c	Preview local picture
MINOR_LOCAL_SET_GATE_PARM_CFG	0x200d	Set local exit&entrance parameters
MINOR_LOCAL_GET_GATE_PARM_CFG	0x200e	Get local exit&entrance parameters
MINOR_LOCAL_SET_DATAUPLOAD_PARM_CFG	0x200f	Set local data uploading parameters
MINOR_LOCAL_GET_DATAUPLOAD_PARM_CFG	0x2010	Get local data uploading parameters
MINOR_LOCAL_DEVICE_CONTROL	0x2011	Control local device (Open/close barrier)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_ADD_EXTERNAL_DEVICE_INFO	0x2012	Add peripheral information (local)
MINOR_LOCAL_MOD_EXTERNAL_DEVICE_INFO	0x2013	Edit peripheral information (local)
MINOR_LOCAL_DEL_EXTERNAL_DEVICE_INFO	0x2014	Delete peripheral information (local)
MINOR_LOCAL_FIND_EXTERNAL_DEVICE_INFO	0x2015	Search peripheral information (local)
MINOR_LOCAL_ADD_CHARGE_RULE	0x2016	Add parking rule (local)
MINOR_LOCAL_MOD_CHARGE_RULE	0x2017	Edit parking rule (local)
MINOR_LOCAL_DEL_CHARGE_RULE	0x2018	Delete parking rule (local)
MINOR_LOCAL_FIND_CHARGE_RULE	0x2019	Search parking rule (local)
MINOR_LOCAL_COUNT_NORMAL_CURRENTINFO	0x2020	Normal passing information statistics (local)
MINOR_LOCAL_EXPORT_NORMAL_CURRENTINFO_REPORT	0x2021	Export normal passing information report (local)
MINOR_LOCAL_COUNT_ABNORMAL_CURRENTINFO	0x2022	Abnormal passing information statistics (local)
MINOR_LOCAL_EXPORT_ABNORMAL_CURRENTINFO_REPORT	0x2023	Export abnormal passing information report (local)
MINOR_LOCAL_COUNT_PEDESTRIAN_CURRENTINFO	0x2024	Pedestrian passing information statistics (local)
MINOR_LOCAL_EXPORT_PEDESTRIAN_CURRENTINFO_REPORT	0x2025	Export pedestrian passing information report (local)
MINOR_LOCAL_FIND_CAR_CHARGEINFO	0x2026	Search vehicle passing fee information (local)
MINOR_LOCAL_COUNT_CAR_CHARGEINFO	0x2027	Vehicle passing fee information statistics (local)
MINOR_LOCAL_EXPORT_CAR_CHARGEINFO_REPORT	0x2028	Export vehicle passing fee information report (local)
MINOR_LOCAL_FIND_SHIFTINFO	0x2029	Search shift information (local)
MINOR_LOCAL_FIND_CARDINFO	0x2030	Search card information (local)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_ADD_RELIEF_RULE	0x2031	Add discount rule (local)
MINOR_LOCAL_MOD_RELIEF_RULE	0x2032	Edit discount rule (local)
MINOR_LOCAL_DEL_RELIEF_RULE	0x2033	Delete discount rule (local)
MINOR_LOCAL_FIND_RELIEF_RULE	0x2034	Search discount rule (local)
MINOR_LOCAL_GET_ENDETCFG	0x2035	Get configuration parameters for entrance&exit station offline detection (local)
MINOR_LOCAL_SET_ENDETCFG	0x2036	Set configuration parameters for entrance&exit station offline detection (local)
MINOR_LOCAL_SET_ENDEV_ISSUEDDATA	0x2037	Set card applying information for entrance&exit station (local)
MINOR_LOCAL_DEL_ENDEV_ISSUEDDATA	0x2038	Clear card applying information for entrance&exit station (local)
MINOR_REMOTE_DEVICE_CONTROL	0x2101	Remote device control
MINOR_REMOTE_SET_GATE_PARM_CFG	0x2102	Set entrance&exit parameters for remote configuration
MINOR_REMOTE_GET_GATE_PARM_CFG	0x2103	Get entrance&exit parameters for remote configuration
MINOR_REMOTE_SET_DATAUPLOAD_PARM_CFG	0x2104	Set data uploading parameters for remote configuration
MINOR_REMOTE_GET_DATAUPLOAD_PARM_CFG	0x2105	Get data uploading parameters for remote configuration
MINOR_REMOTE_GET_BASE_INFO	0x2106	Get terminal basic information (remote)
MINOR_REMOTE_GET_OVERLAP_CFG	0x2107	Get text overlay parameters (remote)
MINOR_REMOTE_SET_OVERLAP_CFG	0x2108	Set text overlay parameters (remote)
MINOR_REMOTE_GET_ROAD_INFO	0x2109	Get crossing information (remote)
MINOR_REMOTE_START_TRANSCHAN	0x210a	Build data synchronizing server (remote)
MINOR_REMOTE_GET_ECTWORKSTATE	0x210b	Get entrance&exit terminal working status (remote)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_REMOTE_GET_ECTCHANINFO	0x210c	Get entrance&exit terminal channel status (remote)
MINOR_REMOTE_ADD_EXTERNAL_DEVICE_INFO	0x210d	Add peripheral information (remote)
MINOR_REMOTE_MOD_EXTERNAL_DEVICE_INFO	0x210e	Edit peripheral information (remote)
MINOR_REMOTE_GET_ENDETCFG	0x210f	Get configuration parameters for entrance&exit station offline detection (remote)
MINOR_REMOTE_SET_ENDETCFG	0x2110	Set configuration parameters for entrance&exit station offline detection (remote)
MINOR_REMOTE_ENDEV_ISSUEDDATA	0x2111	Set card applying information for entrance&exit station (remote)
MINOR_REMOTE_DEL_ENDEV_ISSUEDDATA	0x2112	Clear card applying information for entrance&exit station (remote)
MINOR_REMOTE_ON_CTRL_LAMP	0x2115	Enable remote control parking indicator
MINOR_REMOTE_OFF_CTRL_LAMP	0x2116	Disable remote control parking indicator
MINOR_SET_VOICE_LEVEL_PARAM	0x2117	Set volume
MINOR_SET_VOICE_INTERCOM_PARAM	0x2118	Set recording volume
MINOR_SET_INTELLIGENT_PARAM	0x2119	VCA configuration
MINOR_LOCAL_SET_RAID_SPEED	0x211a	Set raid speed (local)
MINOR_REMOTE_SET_RAID_SPEED	0x211b	Set raid speed (remote)
MINOR_REMOTE_CREATE_STORAGE_POOL	0x211c	Add storage pool (remote)
MINOR_REMOTE_DEL_STORAGE_POOL	0x211d	Delete storage pool (remote)
MINOR_REMOTE_DEL_PICTURE	0x2120	Delete picture data (remote)
MINOR_REMOTE_DEL_RECORD	0x2121	Delete recording data (remote)
MINOR_REMOTE_CLOUD_ENABLE	0x2123	Enable cloud storage (remote)

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_REMOTE_CLOUD_DISABLE	0x2124	Disable cloud storage (remote)
MINOR_REMOTE_CLOUD MODIFY_PARAM	0x2125	Edit cloud storage pool parameters (remote)
MINOR_REMOTE_CLOUD MODIFY_VOLUME	0x2126	Edit cloud storage pool capacity (remote)
MINOR_REMOTE_CREATE_MOD_VIEWLIB_SPACE	0x2200	Create/edit image library space (remote)
MINOR_REMOTE_DELETE_VIEWLIB_FILE	0x2201	Delete image library file (remote)
MINOR_REMOTE_DOWNLOAD_VIEWLIB_FILE	0x2202	Download image library file(s) (remote)
MINOR_REMOTE_UPLOAD_VIEWLIB_FILE	0x2203	Upload image library file(s) (remote)
MINOR_LOCAL_CREATE_MOD_VIEWLIB_SPACE	0x2204	Create/edit image library space (local)
MINOR_REMOTE_CONFERENCE_CONFIG	0x2501	MCU meeting configuration
MINOR_REMOTE_TERMINAL_CONFIG	0x2502	MCU terminal configuration
MINOR_REMOTE_GROUP_CONFIG	0x2503	MCU group configuration
MINOR_REMOTE_CONFERENCE_CTRL	0x2504	MCU meeting control
MINOR_REMOTE_TERMINAL_CTRL	0x2505	MCU terminal control
MINOR_LOCAL_RESET_LOGIN_PASSWORD	0x2600	Reset password for admin user (local)
MINOR_REMOTE_RESET_LOGIN_PASSWORD	0x2601	Reset password for admin user (remote)
MINOR_LOCAL_FACE_BASE_CREATE	0x2602	Create local face picture library
MINOR_REMOTE_FACE_BASE_CREATE	0x2603	Create remote face picture library
MINOR_LOCAL_FACE_BASE MODIFY	0x2604	Edit local face picture library
MINOR_REMOTE_FACE_BASE MODIFY	0x2605	Edit remote face picture library
MINOR_LOCAL_FACE_BASE_DELETE	0x2606	Delete local face picture library
MINOR_REMOTE_FACE_BASE_DELETE	0x2607	Delete remote face picture library

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_LOCAL_FACE_DATA_APPEND	0x2608	Add face data locally
MINOR_REMOTE_FACE_DATA_APPEND	0x2609	Add face data remotely
MINOR_LOCAL_FACE_DATA_SEARCH	0x2610	Search local face comparison data
MINOR_REMOTE_FACE_DATA_SEARCH	0x2611	Search remote face comparison data
MINOR_LOCAL_FACE_DATA_ANALYSIS	0x2612	Analysis picture locally
MINOR_REMOTE_FACE_DATA_ANALYSIS	0x2613	Analysis picture remotely
MINOR_LOCAL_FACE_DATA_EDIT	0x2614	Edit local face data
MINOR_REMOTE_FACE_DATA_EDIT	0x2615	Edit remote face data
MINOR_LOCAL_FACE_DATA_DELETE	0x2616	Delete local face data
MINOR_REMOTE_FACE_DATA_DELET	0x2617	Delete remote face data
MINOR_LOCAL_VCA_ANALYSIS_CFG	0x2618	Set local intelligent analysis
MINOR_REMOTE_VCA_ANALYSIS_CFG	0x2619	Set remote intelligent analysis
MINOR_LOCAL_FACE_BASE_IMPORT	0x261a	Import face picture library locally
MINOR_LOCAL_FACE_BASE_EXPORT	0x261b	Export face picture library locally
MINOR_LOCAL_ADDRESS_FILTER_CONFIG	0x2633	Local address filter configuration
MINOR_REMOTE_ADDRESS_FILTER_CONFIG	0x2634	Remote address filter configuration
MINOR_LOCAL_SSD_UPGRADE_START	0x2639	Upgrade of local SSD file system started
MINOR_LOCAL_SSD_UPGRADE_STOP	0x2640	Upgrade of local SSD file system ended
MINOR_REMOTE_SSD_UPGRADE_START	0x2641	Upgrade of remote SSD file system started
MINOR_REMOTE_SSD_UPGRADE_STOP	0x2642	Upgrade of remote SSD file system ended
MINOR_LOCAL_AUTO_SWITCH_CONFIG	0x2647	Configure auto power on or off locally

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_REMOTE_AUTO_SWITCH_CONFIG	0x2648	Configure auto power on or off remotely
MINOR_REMOTE_AI_MODEL_ADD	0x2650	Add model package
MINOR_REMOTE_AI_MODEL_QUERY	0x2651	Search for model package
MINOR_REMOTE_AI_MODEL_DELETE	0x2652	Delete model package
MINOR_REMOTE_AI_MODEL_UPDATE	0x2653	Update model package
MINOR_REMOTE_AI_PICTURE_POLLING_TASK_ADD	0x2654	Add picture polling task
MINOR_REMOTE_AI_PICTURE_POLLING_TASK_QUERY	0x2655	Search for picture polling task
MINOR_REMOTE_AI_PICTURE_POLLING_TASK_DELETE	0x2656	Delete picture polling task
MINOR_REMOTE_AI_PICTURE_POLLING_TASK MODIFY	0x2657	Edit picture polling task
MINOR_REMOTE_AI_VIDEO_POLLING_TASK_ADD	0x2658	Add video polling task
MINOR_REMOTE_AI_VIDEO_POLLING_TASK_QUERY	0x2659	Search for video polling task
MINOR_REMOTE_AI_VIDEO_POLLING_TASK_DELETE	0x265A	Delete video polling task
MINOR_REMOTE_AI_VIDEO_POLLING_TASK MODIFY	0x265B	Edit video polling task
MINOR_REMOTE_AI_PICTURE_TASK_ADD	0x265C	Add picture task
MINOR_REMOTE_AI_PICTURE_TASK_QUERY	0x265D	Search for picture task
MINOR_REMOTE_AI_PICTURE_TASK_DELETE	0x265E	Delete picture task
MINOR_REMOTE_AI_PICTURE_TASK MODIFY	0x265F	Edit picture task
MINOR_REMOTE_AI_VIDEO_TASK_ADD	0x2660	Add video task

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_REMOTE_AI_VIDEO_TASK_QUERY	0x2661	Search for video task
MINOR_REMOTE_AI_VIDEO_TASK_DELETE	0x2662	Delete video task
MINOR_REMOTE_AI_VIDEO_TASK MODIFY	0x2663	Edit video task
MINOR_LOCAL_SSD_OPERATE_START	0x2705	Local SSD operation started
MINOR_LOCAL_SSD_OPERATE_STOP	0x2706	Local SSD operation ended
MINOR_REMOTE_SSD_OPERATE_START	0x2707	Remote SSD operation started
MINOR_REMOTE_SSD_OPERATE_STOP	0x2708	Remote SSD operation ended
MINOR_LOCAL_EZVIZ_OPERATION	0x2671	Local EZVIZ operations
MINOR_REMOTE_EZVIZ_OPERATION	0x2672	Remote EZVIZ operations
MINOR_LOCAL_PARA_FACTORY_DEFAULT	0x3002	Restore to default settings locally
MINOR_REMOTE_PARA_FACTORY_DEFAULT	0x3003	Restore to default settings remotely
MIMOR_REMOTE_DELETE_ALL_VERIFYORCAP_PICS	0x3004	Delete all authenticated or captured face pictures remotely
MIMOR_LOCAL_DELETE_ALL_VERIFYORCAP_PICS	0x3005	Delete all authenticated or captured face pictures locally
MIMOR_REMOTE_DELETE_EVENTS_AT_SPECETIME	0x3006	Delete events by specified time remotely
MIMOR_LOCAL_DELETE_EVENTS_AT_SPECETIME	0x3007	Delete events by specified time locally
MIMOR_REMOTE_OPEN_SUMMER_TIME	0x3008	Enable DST (Daylight Saving Time) remotely
MIMOR_LOCAL_OPEN_SUMMER_TIME	0x3009	Enable DST (Daylight Saving Time) locally
MIMOR_REMOTE_CLOSE_SUMMER_TIME	0x3010	Disable DST (Daylight Saving Time) remotely

Log Minor Type	Value	Description
MIMOR_LOCAL_CLOSE_SUMMER_TIME	0x3011	Disable DST (Daylight Saving Time) locally
MIMOR_REMOTE_EZVIZ_UNBIND	0x3012	Unbind from EZVIZ cloud remotely
MIMOR_LOCAL_EZVIZ_UNBIND	0x3013	Unbind from EZVIZ cloud locally
MIMOR_ENTER_LOCALUI_BACKGROUND	0x3014	Enter UI background
MIMOR_REMOTE_DELETE_FACEBASEMAP	0x3015	Delete registered face pictures remotely
MIMOR_LOCAL_DELETE_FACEBASEMAP	0x3016	Delete registered face pictures locally
MINOR_SSH_ENABLE	0xc55	SSH switch configuration

## MAJOR INFORMATION

**Table B-4 Minor Types of Additional Information Log**

Log Minor Type	Value	Description
MINOR_HDD_INFO	0xa1	HDD Information
MINOR_SMART_INFO	0xa2	S.M.A.R.T Information
MINOR_REC_START	0xa3	Start recording
MINOR_REC_STOP	0xa4	Stop recording
MINOR_REC_OVERDUE	0xa5	Delete expired video files
MINOR_LINK_START	0xa6	Connect front-end device
MINOR_LINK_STOP	0xa7	Disconnect front-end device
MINOR_NET_DISK_INFO	0xa8	Network HDD information
MINOR_RAID_INFO	0xa9	raid related information
MINOR_RUN_STATUS_INFO	0xaa	System running status information
MINOR_SPARE_START_BACKUP	0xab	Hot spare system starts backing up working device
MINOR_SPARE_STOP_BACKUP	0xac	Hot spare system stops backing up working device

Log Minor Type	Value	Description
MINOR_SPARE_CLIENT_INFO	0xad	Hot spare customer device information
MINOR_ANR_RECORD_START	0xae	Start ANR recording
MINOR_ANR_RECORD_END	0xaf	Stop ANR recording
MINOR_ANR_ADD_TIME_QUANTUM	0xb0	Add ANR time period
MINOR_ANR_DEL_TIME_QUANTUM	0xb1	Delete ANR time period
MINOR_PIC_REC_START	0xb3	Start capturing
MINOR_PIC_REC_STOP	0xb4	Stop Capturing
MINOR_PIC_REC_OVERDUE	0xb5	Delete expired picture
MINOR_CLIENT_LOGIN	0xb6	Logging in to server completed
MINOR_CLIENT_RELOGIN	0xb7	Log in to server again
MINOR_CLIENT_LOGOUT	0xb8	Exiting server completed
MINOR_CLIENT_SYNC_START	0xb9	Start Synchronous Recording
MINOR_CLIENT_SYNC_STOP	0xba	Stop Synchronous Recording
MINOR_CLIENT_SYNC_SUCC	0xbb	Synchronous Recording Completed
MINOR_CLIENT_SYNC_EXCP	0xbc	Synchronous recording exception
MINOR_GLOBAL_RECORD_ERR_INFO	0xbd	Global Error Information
MINOR_BUFFER_STATE	0xbe	Buffer Status Log File
MINOR_DISK_ERRORINFO_V2	0xbf	HDD Error Details V2
MINOR_UNLOCK_RECORD	0xc3	Lock Record
MINOR_VIS_ALARM	0xc4	Zone Alarm
MINOR_TALK_RECORD	0xc5	Calling Record
MINOR_ACCESSORIES_MESSAGE	0xc6	Accessories Information
MINOR_IPC_CONNECT	0xc8	Network connection information
MINOR_INTELLIGENT_BOARD_STATUS	0xc9	Intelligent board status
MINOR_IPC_CONNECT_STATUS	0xca	Network camera connection status
MINOR_EZVIZ_OPERATION	0xcc	EZVIZ Running Status
MINOR_CLUSTER_DEVICE_ONLINE	0xcd	Cluster device is online

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_CLUSTER_MGR_SERVICE_STARTUP	0xce	Cluster management service is enabled
MINOR_CLUSTER_BUSINESS_TRANSFER	0xcf	Cluster migration
MINOR_CLUSTER_STATUS	0xd0	Cluster status information
MINOR_CLUSTER_CS_STATUS	0xd1	Sending device status to CM failed. Record the IP address of CS and CM.
MINOR_CLUSTER_CM_STATUS	0xd2	CM status switching.
MINOR_DOUBLE_VERIFICATION_PASS	0xd5	Double verification completed
MINOR_HD_FORMAT_START	0xd8	Formatting HDD started.
MINOR_HD_FORMAT_STOP	0xd9	Formatting HDD stopped.
MINOR_802_1X_AUTH_SUCC	0x320	802.1x authentication succeeded.
MINOR_802_1X_AUTH_FAIL	0x321	802.1x authentication failed.
MINOR_LIVE_DETECT_OPEN	0x400	Enabled face anti-spoofing detection
MINOR_LIVE_DETECT_CLOSE	0x401	Disabled face anti-spoofing detection
MINOR_CLEAR_DATA_COLLECTION	0x402	Cleared collected data
MINOR_DELETE_DATA_COLLECTION	0x403	Deleted collected data
MINOR_EXPORT_DATA_COLLECTION	0x404	Exported collected data
MINOR_CARD_LEN_CONFIG	0x405	Configured card number size
MINOR_DATA_BASE_INIT_FAILED	0x406	Initializing database failed
MINOR_DATA_BASE_PATCH_UPDATE	0x407	Upgraded database patch
MINOR_PSAM_CARD_INSERT	0x408	Inserted PSAM card
MINOR_PSAM_CARD_REMOVE	0x409	Pulled out PSAM card
MINOR_HARD_FAULT_REBOOT	0x40a	Reboot as hardware exception
MINOR_PSAM_CARD_OCP	0x40b	Overflow protection of PSAM card
MINOR_STACK_OVERFLOW	0x40c	Stack overflow
MINOR_PARM_CFG	0x40d	Parameter configuration
MINOR_CLR_USER	0x40e	Clear all users
MINOR_CLR_CARD	0x40f	Clear all cards

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_CLR_FINGER_BY_READER	0x410	Clear all fingerprints by fingerprint and card reader
MINOR_CLR_FINGER_BY_CARD	0x411	Clear all fingerprints by card No.
MINOR_CLR_FINGER_BY_EMPLOYEE_ON	0x412	Clear all fingerprints by employee ID
MINOR_DEL_FINGER	0x413	Delete a fingerprint
MINOR_CLR_WEEK_PLAN	0x414	Clear week schedules of access permission control
MINOR_SET_WEEK_PLAN	0x415	Set the week schedule of access permission control
MINOR_SET_HOLIDAY_PLAN	0x416	Set the holiday schedule of access permission control
MINOR_CLR_HOLIDAY_PLAN	0x417	Clear holiday schedules of access permission control
MINOR_SET_HOLIDAY_GROUP	0x418	Set the holiday group of access permission control schedule
MINOR_CLR_HOLIDAY_GROUP	0x419	Clear holiday groups of access permission control schedule
MINOR_CLR_TEMPLATE_PLAN	0x41a	Clear access permission control schedules
MINOR_SET_TEMPLATE_PLAN	0x41b	Set the access permission control schedule
MINOR_ADD_CARD	0x41c	Add a card
MINOR_MOD_CARD	0x41d	Edit a card
MINOR_ADD_FINGER_BY_CARD	0x41e	Add a fingerprint by card No.
MINOR_ADD_FINGER_BY_EMPLOYEE_NO	0x41f	Add a fingerprint by employee ID
MINOR_MOD_FINGER_BY_CARD	0x420	Edit a fingerprint by card No.
MINOR_MOD_FINGER_BY_EMPLOYEE_NO	0x421	Edit a fingerprint by employee ID
MINOR_IMPORT_USER_LIST	0x422	Imported user list (offline data collection)

Log Minor Type	Value	Description
MINOR_USB_LOGIN	0x423	Log in via USB
MINOR_USB_LOGOUT	0x424	Log out via USB
MINOR_ISAPI_HTTP_LOGIN	0x425	Log in via text protocol (HTTP)
MINOR_ISAPI_HTTP_LOGOUT	0x426	Log out via text protocol (HTTP)
MINOR_ISAPI_HTTPS_LOGIN	0x427	Log in via text protocol (HTTPS)
MINOR_ISAPI_HTTPS_LOGOUT	0x428	Log out via text protocol (HTTPS)
MINOR_ISUP_ONLINE	0x429	ISUP online
MINOR_ISUP_OFFLINE	0x42a	ISUP offline
MINOR_FP_ISSUE_REC	0x42b	Issuing record of card containing fingerprint information
MINOR_FACE_ISSUE_REC	0x42c	Issuing record of card containing face picture information
MINOR_ADD_USER_INFO	0x432	Added person information (access control permission)
MINOR MODIFY_USER_INFO	0x433	Edit person information (access control permission)
MINOR_CLR_USER_INFO	0x434	Delete person information by employee No. (access control permission)
MINOR_CLR_CARD_BY_CARD_OR_EMPLOYEE	0x435	Delete cards by card No. or employee No.
MINOR_WIRELESS_RUNNING_STATUS	0xd6	Wireless network running status

## MAJOR\_EVENT

Table B-5 Minor Types of Event Log

Log Minor Type	Value	Description
MINOR_LEGAL_CARD_PASS	0x01	Legal Card Authenticated
MINOR_CARD_AND_PSW_PASS	0x02	Card and Password Authenticated
MINOR_CARD_AND_PSW_FAIL	0x03	Card and password authentication failed.

<b>Log Minor Type</b>	<b>Value</b>	<b>Description</b>
MINOR_CARD_AND_PSW_TIMEOUT	0x04	Card and password authentication timed out.
MINOR_CARD_AND_PSW_OVER_TIME	0x05	Card and password timed out.
MINOR_CARD_NO_RIGHT	0x06	No Permission
MINOR_CARD_INVALID_PERIOD	0x07	Invalid Duration
MINOR_CARD_OUT_OF_DATE	0x08	Expired Card
MINOR_INVALID_CARD	0x09	No card No.
MINOR_ANTI_SNEAK_FAIL	0x0a	Anti-passing back authentication failed.
MINOR_INTERLOCK_DOOR_NOT_CLOSE	0x0b	Interlocking Door Not Closed
MINOR_NOT_BELONG_MULTI_GROUP	0x0c	The card does not belong to multiple authentication group.
MINOR_INVALID_MULTI_VERIFY_PERIOD	0x0d	The card is not in the multiple authentication duration.
MINOR_MULTI_VERIFY_SUPER_RIGHT_FAIL	0x0e	Multiple Authentication: Super Permission Authentication Failed
MINOR_MULTI_VERIFY_REMOTE_RIGHT_FAIL	0x0f	Multiple Authentication: Remote Authentication Failed
MINOR_MULTI_VERIFY_SUCCESS	0x10	Pass Multiple Authentication
MINOR_LEADER_CARD_OPEN_BEGIN	0x11	Open Door with First Card Started
MINOR_LEADER_CARD_OPEN_END	0x12	Open Door with First Card Stopped
MINOR_ALWAYS_OPEN_BEGIN	0x13	Remain Open Started
MINOR_ALWAYS_OPEN_END	0x14	Remain Open Stopped
MINOR_LOCK_OPEN	0x15	Unlock Door
MINOR_LOCK_CLOSE	0x16	Lock Door
MINOR_DOOR_BUTTON_PRESS	0x17	Press Door Button
MINOR_DOOR_BUTTON_RELEASE	0x18	Release Door Button
MINOR_DOOR_OPEN_NORMAL	0x19	Normal Open (Door Magnetic)
MINOR_DOOR_CLOSE_NORMAL	0x1a	Normal Closed (Door Magnetic)

Log Minor Type	Value	Description
MINOR_DOOR_OPEN_ABNORMAL	0x1b	Abnormal Open (Door Magnetic)
MINOR_DOOR_OPEN_TIMEOUT	0x1c	Open Door Timeout (Door Magnetic)
MINOR_ALARMOUT_ON	0x1d	Alarm Output On
MINOR_ALARMOUT_OFF	0x1e	Alarm Output Off
MINOR_ALWAYS_CLOSE_BEGIN	0x1f	Remain Open Started
MINOR_ALWAYS_CLOSE_END	0x20	Remain Open Stopped
MINOR_MULTI_VERIFY_NEED_REMOTE_OPEN	0x21	Multiple Authentication: Remote Open Door
MINOR_MULTI_VERIFY_SUPERPASSWD_VERIFY_SUCCESS	0x22	Multiple Authentication: Super Password Authentication Passed
MINOR_MULTI_VERIFY_REPEAT_VERIFY	0x23	Multiple Authentication: Repeat Authentication
MINOR_MULTI_VERIFY_TIMEOUT	0x24	Multiple Authentication: Repeat Authentication Event

## Appendix D. Device Network SDK Errors

The errors that may occur during the device network SDK integration are listed here for reference. You can search for the error descriptions according to the error codes or names returned by a specific API (NET\_DVR\_GetLastError or NET\_DVR\_GetErrorMsg).

### General Errors

Error Name	Error Code	Error Description
NET_DVR_NOERROR	0	No error.
NET_DVR_PASSWORD_ERROR	1	Incorrect user name or password.
NET_DVR_NOENOUGHPRI	2	No permission.
NET_DVR_NOINIT	3	Uninitialized.
NET_DVR_CHANNEL_ERROR	4	Incorrect channel No.
NET_DVR_OVER_MAXLINK	5	No more device can be connected.
NET_DVR_VERSIONNOMATCH	6	Version mismatches.
NET_DVR_NETWORK_FAIL_CONNECT	7	Connecting to device failed. The device is offline or network connection timed out.
NET_DVR_NETWORK_SEND_ERROR	8	Sending data to device failed.
NET_DVR_NETWORK_RECV_ERROR	9	Receiving data from device failed.
NET_DVR_NETWORK_RECV_TIMEOUT	10	Receiving data from device timed out.
NET_DVR_NETWORK_ERRORDATA	11	The data sent to the device is illegal, or the data received from the device error. E.g. The input data is not supported by the device for remote configuration.
NET_DVR_ORDER_ERROR	12	API calling order error.
NET_DVR_OPERNOOPERMIT	13	No permission for this operation.
NET_DVR_COMMANDTIMEOUT	14	Executing device command timed out.
NET_DVR_ERRORSERIALPORT	15	Incorrect serial port No. The specified serial port does not exist.

Error Name	Error Code	Error Description
NET_DVR_ERRORALARMPORT	16	Alarm port No. error. The alarm input or output port of the specified device does not exist.
NET_DVR_PARAMETER_ERROR	17	Incorrect parameter. The input or output parameters of the SDK API is empty, or the parameter value or format is invalid.
NET_DVR_CHAN_EXCEPTION	18	Device channel is in exception status.
NET_DVR_NODISK	19	No HDD in the device.
NET_DVR_ERRORDISKNUM	20	Incorrect HDD No.
NET_DVR_DISK_FULL	21	HDD full.
NET_DVR_DISK_ERROR	22	HDD error.
NET_DVR_NOSUPPORT	23	Device does not support this function.
NET_DVR_BUSY	24	Device is busy.
NET_DVR MODIFY_FAIL	25	Failed to edit device parameters.
NET_DVR_PASSWORD_FORMAT_ERROR	26	Invalid password format.
NET_DVR_DISK_FORMATING	27	HDD is formatting. Failed to startup.
NET_DVR_DVRNORESOURCE	28	Insufficient device resources.
NET_DVR_DVROPRATEFAILED	29	Device operation failed.
NET_DVR_OPENHOSTSOUND_FAIL	30	Failed to collect local audio data or open audio output during two-way audio and broadcast.
NET_DVR_DVRVOICEOPENED	31	Two-way audio channel is occupied.
NET_DVR_TIMEINPUTERROR	32	Incorrect time input.
NET_DVR_NOSPECFILE	33	No video file for playback.
NET_DVR_CREATEFILE_ERROR	34	Failed to create a file during local recording, saving picture, getting configuration file or downloading video file remotely.
NET_DVR_FILEOPENFAIL	35	Failed to open a file. The file does not exist or directory error.

Error Name	Error Code	Error Description
NET_DVR_OPERNOTFINISH	36	Operation conflicted.
NET_DVR_GETPLAYTIMEFAIL	37	Failed to get the current played time.
NET_DVR_PLAYFAIL	38	Failed to play.
NET_DVR_FILEFORMAT_ERROR	39	Invalid file format.
NET_DVR_DIR_ERROR	40	File directory error.
NET_DVR_ALLOC_RESOURCE_ERROR	41	Allocating resources failed.
NET_DVR_AUDIO_MODE_ERROR	42	Invalid sound card mode error. The opened sound play mode and configured mode mismatched.
NET_DVR_NOENOUGH_BUF	43	Insufficient buffer for receiving data or saving picture.
NET_DVR_CREATESOCKET_ERROR	44	Failed to create SOCKET.
NET_DVR_SETSOCKET_ERROR	45	Failed to set SOCKET.
NET_DVR_MAX_NUM	46	No more registrations and live views can be connected.
NET_DVR_USERNOTEXIST	47	The user does not exist. The user ID is logged out or unavailable.
NET_DVR_WRITEFLASHERROR	48	Writing FLASH error during device upgrade.
NET_DVR_UPGRADEFAIL	49	Failed to upgrade device. Network problem or language mismatches.
NET_DVR_CARDHAVEINIT	50	The decoding card is already initialized.
NET_DVR_PLAYERFAILED	51	Failed to call the function of player SDK.
NET_DVR_MAX_USERNUM	52	No more users can log in to.
NET_DVR_GETLOCALIPANDMACFAIL	53	Failed to get the IP address or physical address of local PC.
NET_DVR_NOENCODEING	54	The decoding function of this channel is not enabled.
NET_DVR_IPMISMATCH	55	IP address mismatches.

Error Name	Error Code	Error Description
NET_DVR_MACMISMATCH	56	MAC address mismatches.
NET_DVR_UPGRADELANGMISMATCH	57	The language of upgrade file mismatches.
NET_DVR_MAX_PLAYERPORT	58	No more channels can be started to play.
NET_DVR_NOSPACEBACKUP	59	Insufficient space to back up file.
NET_DVR_NODEVICEBACKUP	60	No backup device found.
NET_DVR_PICTURE_BITS_ERROR	61	Picture pixel bit mismatches. Only 24 bits are allowed.
NET_DVR_PICTURE_DIMENSION_ERROR	62	Too large picture. The height*width should be less than 128x256.
NET_DVR_PICTURE_SIZ_ERROR	63	Too large picture. The picture size should be smaller than 100K.
NET_DVR_LOADPLAYERSDKFAILED	64	Failed to load the player(PlayCtrl.dll, SuperRender.dll, AudioRender.dll) to the current directory.
NET_DVR_LOADPLAYERSDKPROC_ERROR	65	Failed to find the function in player SDK.
NET_DVR_LOADDSSDKFAILED	66	Failed to load the DS SDK to the current directory.
NET_DVR_LOADDSSDKPROC_ERROR	67	Failed to find the function in the DS SDK.
NET_DVR_DSSDK_ERROR	68	Failed to call the API in the hardware decoding library.
NET_DVR_VOICEMONOPOLIZE	69	The sound card is exclusive.
NET_DVR_JOINMULTICASTFAILED	70	Failed to join to multicast group.
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	Socket disconnected. Network disconnected or the destination is unreachable.

Error Name	Error Code	Error Description
NET_DVR_USERID_ISUSING	74	Operation is executing. Failed to log out.
NET_DVR_SOCKETLISTEN_ERROR	75	Failed to listen.
NET_DVR_PROGRAM_EXCEPTION	76	Program exception.
NET_DVR_WRITEFILE_FAILED	77	Failed to write file during local recording, downloading file remotely or saving picture.
NET_DVR_FORMAT_READONLY	78	The HDD is read-only. Formatting is forbidden.
NET_DVR_WITHSAMEUSERNAME	79	The user name already exists.
NET_DVR_DEVICETYPE_ERROR	80	Device model mismatches when importing parameters.
NET_DVR_LANGUAGE_ERROR	81	Language mismatches when importing parameters.
NET_DVR_PARAVERSION_ERROR	82	Software version mismatches when importing parameters.
NET_DVR_IPCHAN_NOTALIVE	83	The external IP channel is offline live view.
NET_DVR_RTSP_SDK_ERROR	84	Failed to load StreamTransClient.dll.
NET_DVR_CONVERT_SDK_ERROR	85	Failed to load SystemTransform.dll.
NET_DVR_IPC_COUNT_OVERFLOW	86	No more IP channels can access to.
NET_DVR_MAX_ADD_NUM	87	No more video tags can be added.
NET_DVR_PARAMMODE_ERROR	88	Invalid parameter mode of image enhancement.
NET_DVR_CODESPITTER_OFFLINE	89	Code distributer is offline.
NET_DVR_BACKUP COPYING	90	Device is backing up.
NET_DVR_CHAN_NOTSUPPORT	91	This operation is not supported by the channel.
NET_DVR_CALLINEINVALID	92	The height line is too concentrated, or the length line is not inclined enough.

Error Name	Error Code	Error Description
NET_DVR_CALCANCELCONFLICT	93	Cancel calibration conflict, if the rule and global actual size filter are configured.
NET_DVR_CALPOINTOUTRANGE	94	The calibration point is out of limitation.
NET_DVR_FILTERRECTINVALID	95	The size filter does not meet the requirement.
NET_DVR_DDNS_DEVOFFLINE	96	Device has not registered to DDNS.
NET_DVR_DDNS_INTER_ERROR	97	DDNS internal error.
NET_DVR_FUNCTION_NOT_SUPPORT_OS	98	This function is not supported by this Operating system.
NET_DVR_DEC_CHAN_REBIND	99	Decoding channel binding display output is limited.
NET_DVR_INTERCOM_SDK_ERROR	100	Failed to load the two-way audio SDK of the current directory.
NET_DVR_NO_CURRENT_UPDATEFILE	101	No correct upgrade packet.
NET_DVR_USER_NOT_SUCC_LOGIN	102	Login failed.
NET_DVR_USE_LOG_SWITCH_FILE	103	The log switch file is under using.
NET_DVR_POOL_PORT_EXHAUST	104	No port can be bound in the port pool.
NET_DVR_PACKET_TYPE_NOT_SUPPORT	105	Incorrect stream packaging format.
NET_DVR_IPPARA_IPID_ERROR	106	Incorrect IPID for IP access configuration.
NET_DVR_LOAD_HCPREVIEW_SDK_ERROR	107	Failed to load the live view component.
NET_DVR_LOAD_HCVOICETALK_SDK_ERROR	108	Failed to load the audio component.
NET_DVR_LOAD_HCALARM_SDK_ERROR	109	Failed to load the alarm component.
NET_DVR_LOAD_HCPLAYBACK_SDK_ERROR	110	Failed to load the playback component.

Error Name	Error Code	Error Description
NET_DVR_LOAD_HCDISPLAY_SDK_ERROR	111	Failed to load the display component.
NET_DVR_LOAD_HCINDUSTRY_SDK_ERROR	112	Failed to load application component.
NET_DVR_LOAD_HCGENERALCFGMGR_SDK_ERROR	113	Failed to load the general configuration management component.
NET_DVR_CORE_VER_MISMATCH	121	Component version and core version mismatched when loading the component singly.
NET_DVR_CORE_VER_MISMATCH_HCPREVIEW	122	Live view component version and core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCVOICETALK	123	Audio component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCALARM	124	Alarm component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCPLAYBACK	125	Playback component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCDISPLAY	126	Display component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCINDUSTRY	127	Application component version and the core version mismatched.
NET_DVR_CORE_VER_MISMATCH_HCGENERALCFGMGR	128	General configuration management component version and the core version mismatched.
NET_DVR_COM_VER_MISMATCH_HCPREVIEW	136	Live view component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCVOICETALKy	137	Audio component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCALARM	138	Alarm component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCPLAYBACK	139	Playback component version and SDK version mismatched.

Error Name	Error Code	Error Description
NET_DVR_COM_VER_MISMATCH_HCDISPLAY	140	Display component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCINDUSTRY	141	Application component version and SDK version mismatched.
NET_DVR_COM_VER_MISMATCH_HCGENERALCFGMGR	142	General configuration management component version and SDK version mismatched.
NET_DVR_ALIAS_DUPLICATE	150	Duplicated alias(for HiDDNS configuration).
NET_DVR_USERNAME_NOT_EXIST	152	User name does not exist (error code of network camera and network speed dome with version from 5.1.7 to 5.3.1).
NET_ERR_USERNAME_LOCKED	153	The user name is locked.
NET_DVR_INVALID_USERID	154	Invalid user ID.
NET_DVR_LOW_LOGIN_VERSION	155	The version is too low.
NET_DVR_LOAD_LIBEAY32_DLL_ERROR	156	Failed to load libeay32.dll.
NET_DVR_LOAD_SSLEAY32_DLL_ERROR	157	Failed to load ssleay32.dll.
NET_ERR_LOAD_LIBICONV	158	Failed to load libiconv.dll.
NET_ERR_SSL_CONNECT_FAILED	159	Connecting to SSL failed.
NET_DVR_TEST_SERVER_FAIL_CONNECT	165	Failed to connect to test server.
NET_DVR_NAS_SERVER_INVALID_DIR	166	Failed to load NAS server to the directory, Invalid directory, or incorrect user name and password.
NET_DVR_NAS_SERVER_NOENOUGH_PRI	167	Failed to load NAS server th the directory. No permission.
NET_DVR_EMAIL_SERVER_NOT_CONFIG_DNS	168	The server uses domain name without configuring DNS, the domain name may be invalid.

Error Name	Error Code	Error Description
NET_DVR_EMAIL_SERVER_NOT_CONFIG_GATEWAY	169	No gateway configured. Sending email may be failed.
NET_DVR_TEST_SERVER_PASSWORD_ERROR	170	Incorrect user name or password of test server.
NET_DVR_EMAIL_SERVER_CONNECT_EXCEPTION_WITH_SMTP	171	Interaction exception between device and SMTP server.
NET_DVR_FTP_SERVER_FAIL_CREATE_DIR	172	FTP server creating directory failed.
NET_DVR_FTP_SERVER_NO_WRITE_PIR	173	FTP server has no writing permission.
NET_DVR_IP_CONFLICT	174	IP conflicted.
NET_DVR_INSUFFICIENT_STORAGEPOOL_SPACE	175	Storage pool space is full.
NET_DVR_STORAGEPOOL_INVALID	176	Invalid cloud storage pool. No storage pool configured or incorrect storage pool ID.
NET_DVR_EFFECTIVENESS_REBOOT	177	Restart to take effect.
NET_ERR_ANR_ARMING_EXIST	178	The ANR arming connection already exists( the error will be returned when arming with ANR function if the private SDK protocol arming connection is established).
NET_ERR_UPLOADLINK_EXIST	179	The ANR uploading connection already exists( the error will be returned when EHome protocol and private SDK protocol do not support ANR at the same time).
NET_ERR_INCORRECT_FILE_FORMAT	180	The imported file format is incorrect.
NET_ERR_INCORRECT_FILE_CONTENT	181	The imported file content is incorrect.
NET_ERR_MAX_HRUDP_LINK	182	No more HRUDP can be connected to device.
NET_ERR_MAX_PORT_MULTIPLEX	183	Maximum number of multiplexed ports reaches.
NET_ERR_CREATE_PORT_MULTIPLEX	184	Creating port multiplier failed.

Error Name	Error Code	Error Description
NET_DVR_NONBLOCKING_CAPTURE_NOTSUPPORT	185	Non-blocking picture capture is not supported.
NET_SDK_ERR_FUNCTION_INVALID	186	Invalid function. The asynchronous mode is enabled.
NET_SDK_ERR_MAX_PORT_MULTIPLEX	187	Maximum number of multiplex ports reached.
NET_DVR_INVALID_LINK	188	Link has not been created or the link is invalid.
NET_DVR_NAME_NOT_ONLY	200	This name already exists.
NET_DVR_OVER_MAX_ARRAY	201	The number of RAID reaches the upper-limit.
NET_DVR_OVER_MAX_VD	202	The number of virtual disk reaches the upper-limit.
NET_DVR_VD_SLOT_EXCEED	203	The virtual disk slots are full.
NET_DVR_PD_STATUS_INVALID	204	The physical disk for rebuilding RAID is error.
NET_DVR_PD_BE_DEDICATE_SPARE	205	The physical disk for rebuilding RAID is specified as hot spare.
NET_DVR_PD_NOT_FREE	206	The physical disk for rebuilding RAID is busy.
NET_DVR_CANNOT_MIG2NEWMODE	207	Failed to migrate the current RAID type to the new type.
NET_DVR_MIG_PAUSE	208	Migration is paused.
NET_DVR_MIG_ABOUTED	209	Migration is cancelled.
NET_DVR_EXIST_VD	210	Failed to delete RAID. Virtual disk exists in the RAID.
NET_DVR_TARGET_IN_LD_FUNCTIONAL	211	Target physical disk is a part of the virtual disk and it is working normally.
NET_DVR_HD_IS_ASSIGNED_ALREADY	212	The specified physical disk is allocated as virtual disk.
NET_DVR_INVALID_HD_COUNT	213	The number of physical disks and specified RAID level mismatched.

Error Name	Error Code	Error Description
NET_DVR_LD_IS_FUNCTIONAL	214	The RAID is normal. Failed to rebuild.
NET_DVR_BGA_RUNNING	215	Background task is executing.
NET_DVR_LD_NO_ATAPI	216	Failed to create virtual disk by ATAPI disk.
NET_DVR_MIGRATION_NOT_NEED	217	There is no need to migrate the RAID.
NET_DVR_HD_TYPE_MISMATCH	218	The physical disk type is not allowed.
NET_DVR_NO_LD_IN_DG	219	No virtual disk. Operation failed.
NET_DVR_NO_ROOM_FOR_SPARE	220	Insufficient disk space. Failed to allocate the disk as hot spare.
NET_DVR_SPARE_IS_IN_MULTI_DG	221	The disk is already allocated as the hot spare of one RAID.
NET_DVR_DG_HAS_MISSING_PD	222	No disk in the RAID.
NET_DVR_NAME_EMPTY	223	The name is empty.
NET_DVR_INPUT_PARAM	224	Incorrect input parameters.
NET_DVR_PD_NOT_AVAILABLE	225	The physical disk is not available.
NET_DVR_ARRAY_NOT_AVAILABLE	226	The RAID is not available.
NET_DVR_PD_COUNT	227	Incorrect number of physical disks.
NET_DVR_VD_SMALL	228	Insufficient virtual disk space.
NET_DVR_NO_EXIST	229	Not exist.
NET_DVR_NOT_SUPPORT	230	This operation is not supported.
NET_DVR_NOT_FUNCTIONAL	231	The RAID status is exception.
NET_DVR_DEV_NODE_NOT_FOUND	232	The device node of virtual disk does not exist.
NET_DVR_SLOT_EXCEED	233	No more slots are allowed.
NET_DVR_NO_VD_IN_ARRAY	234	No virtual disk exists in the RAID.
NET_DVR_VD_SLOT_INVALID	235	Invalid virtual disk slot.
NET_DVR_PD_NO_ENOUGH_SPACE	236	Insufficient physical disk space.
NET_DVR_ARRAY_NONFUNCTION	237	Only the RAID in normal status supports to be migrated.

Error Name	Error Code	Error Description
NET_DVR_ARRAY_NO_ENOUGH_SPACE	238	Insufficient RAID space.
NET_DVR_STOPPING_SCANNING_ARRAY	239	Pulling disk out safely or rescanning.
NET_DVR_NOT_SUPPORT_16T	240	Creating RAID with size larger than 16T is not supported.
NET_DVR_ERROR_DEVICE_NOT_ACTIVATED	250	The device is not activated (login failed.)
NET_DVR_ERROR_RISK_PASSWORD	251	Risky password.
NET_DVR_ERROR_DEVICE_HAS_ACTIVATED	252	The device is already activated.
NET_DVR_ID_ERROR	300	The configured ID is invalid.
NET_DVR_POLYGON_ERROR	301	Invalid polygon shape.
NET_DVR_RULE_PARAM_ERROR	302	Invalid rule parameters.
NET_DVR_RULE_CFG_CONFLICT	303	Configured information conflicted.
NET_DVR_CALIBRATE_NOT_READY	304	No calibration information.
NET_DVR_CAMERA_DATA_ERROR	305	Invalid camera parameters.
NET_DVR_CALIBRATE_DATA_UNFIT	306	Invalid inclination angle for calibration.
NET_DVR_CALIBRATE_DATA_CONFLICT	307	Calibration error.
NET_DVR_CALIBRATE_CALC_FAIL	308	Failed to calculate calibration parameter values of camera.
NET_DVR_CALIBRATE_LINE_OUT_RECT	309	The inputted calibration line exceeds the external sample rectangle.
NET_DVR_ENTER_RULE_NOT_READY	310	No region entrance is configured.
NET_DVR_AID_RULE_NO_INCLUDE_LANE	311	No lane configured in the traffic event rule (especially for traffic jam or driving against the traffic).
NET_DVR_LANE_NOT_READY	312	Lane not configured.
NET_DVR_RULE_INCLUDE_TWO_WAY	313	Two different directions are contained in event rule.

Error Name	Error Code	Error Description
NET_DVR_LANE_TPS_RULE_CONFLICT	314	Lane and data rule conflicted.
NET_DVR_NOT_SUPPORT_EVENT_TYPE	315	This event type is not supported.
NET_DVR_LANE_NO_WAY	316	The lane has no direction.
NET_DVR_SIZE_FILTER_ERROR	317	Invalid size of filter frame.
NET_DVR_LIB_FFL_NO_FACE	318	No face picture exists in the image inputted when positioning feature point.
NET_DVR_LIB_FFL_IMG_TOO_SMALL	319	The inputted image is too small when positioning feature point.
NET_DVR_LIB_FD_IMG_NO_FACE	320	No face picture exists in the image inputted when detecting single face picture.
NET_DVR_LIB_FACE_TOO_SMALL	321	Face picture is too small when building model.
NET_DVR_LIB_FACE_QUALITY_TOO_BAD	322	The face picture quality is too poor when building model.
NET_DVR_KEY_PARAM_ERR	323	The configured advanced parameter is incorrect.
NET_DVR_CALIBRATE_DATA_ERR	324	Calibration sample number error, or data value error, or the sample points are beyond the horizontal line.
NET_DVR_CALIBRATE_DISABLE_FAIL	325	Canceling calibration is not allowed for configured rules.
NET_DVR_VCA_LIB_FD_SCALE_OUTRANGE	326	The minimum width and height of maximum filter frame are twice or more larger than the maximum width and height of minimum filter frame.
NET_DVR_LIB_FD_REGION_TOO_LARGE	327	Too large detection region. The maximum region should be 2/3 of the image.
NET_DVR_TRIAL_OVERDUE	328	Trial period is ended.
NET_DVR_CONFIG_FILE_CONFLICT	329	Device type and configuration file conflicted.

Error Name	Error Code	Error Description
NET_DVR_FR_FPL_FAIL	330	Failed to positioning face feature points.
NET_DVR_FR_IQA_FAIL	331	Failed to test face picture quality.
NET_DVR_FR_FEM_FAIL	332	Failed to extract the face feature points.
NET_DVR_FPL_DT_CONF_TOO_LOW	333	The face detection validity is too low when positioning face feature points.
NET_DVR_FPL_CONF_TOO_LOW	334	The validity of feature points positionong is too low.
NET_DVR_E_DATA_SIZE	335	Data size mismatches.
NET_DVR_FR_MODEL_VERSION_ERR	336	Incorrect model version in face model library.
NET_DVR_FR_FD_FAIL	337	Failed to detect face in the face recognition library.
NET_DVR_FA_NORMALIZE_ERR	338	Failed to normalize face attribute.
NET_DVR_DOG_PUSTREAM_NOT_MATCH	339	Dongle type and camera type mismatched.
NET_DVR_DEV_PUSTREAM_NOT_MATCH	340	Camera version mismatches.
NET_DVR_PUSTREAM_ALREADY_EXISTS	341	This camera is already added to other channels of devices.
NET_DVR_SEARCH_CONNECT_FAILED	342	Failed to connect to face retrieval server.
NET_DVR_INSUFFICIENT_DISK_SPACE	343	Insufficient storage space.
NET_DVR_DATABASE_CONNECTION_FAILED	344	Failed to connect to database.
NET_DVR_DATABASE_ADMIN_PW_ERROR	345	Incorrect database user name and password.
NET_DVR_DECODE_YUV	346	Decoding failed.
NET_DVR_IMAGE_RESOLUTION_ERROR	347	Invalid picture resolution

Error Name	Error Code	Error Description
NET_DVR_CHAN_WORKMODE_ERROR	348	Invalid channel working mode.
NET_ERROR_TRUNK_LINE	711	Sub system is configured as the trunk line.
NET_ERROR_MIXED_JOINT	712	Mixed joint is not supported.
NET_ERROR_DISPLAY_SWITCH	713	Switch of display channel is not supported.
NET_ERROR_USED_BY_BIG_SCREEN	714	Decoded resource is occupied by the big screen.
NET_ERROR_USE_OTHER_DEC_RESOURCE	715	Using resources of other sub system is not allowed.
NET_ERROR_SCENE_USING	717	The scene is being used.
NET_ERR_NO_ENOUGH_DEC_RESOURCE	718	Insufficient resources for decoding.
NET_ERR_NO_ENOUGH_FREE_SHOW_RESOURCE	719	Insufficient resources for display.
NET_ERR_NO_ENOUGH_VIDEO_MEMORY	720	Insufficient video storage resources.
NET_ERR_MAX_VIDEO_NUM	721	Insufficient resources for multiple channels.
NET_ERR_WINDOW_COVER_FREE_SHOW_AND_NORMAL	722	Windows cover free display output channel and normal output channel.
NET_ERR_FREE_SHOW_WINDOW_SPLIT	723	Window division is not supported for free display windows.
NET_ERR_INAPPROPRIATE_WINDOW_FREE_SHOW	724	For the windows whose number is not integral multiple of the number of output channels, free display is not supported.
NET_DVR_TRANSPARENT_WINDOW_NOT_SUPPORT_SPLIT	725	For windows whose transparency configuration is enabled, window division is not supported.
NET_DVR_SPLIT_WINDOW_NOT_SUPPORT_TRANSPARENT	726	For windows whose window division is enabled, transparency configuration is not supported.

Error Name	Error Code	Error Description
NET_ERR_TERMINAL_BUSY	780	The terminal busy.
NET_DVR_FUNCTION_RESOURCE_USAGE_ERROR	791	Failed to enable this function. The resources is occupied by other functions.
NET_DVR_DEV_NET_OVERFLOW	800	Network traffic is out of the limitation.
NET_DVR_STATUS_RECORDFILE_WRITING_NOT_LOCK	801	Failed to lock. The video file is recording.
NET_DVR_STATUS_CANT_FORMAT_LITTLE_DISK	802	Failed to format HDD. The HDD space is too small.
NET_SDK_ERR_REMOTE_DISCONNECT	803	Failed to connect to the remote terminal.
NET_SDK_ERR_RD_ADD_RD	804	Spare server cannot be added to spare server.
NET_SDK_ERR_BACKUP_DISK_EXCEPT	805	Backup disk exception.
NET_SDK_ERR_RD_LIMIT	806	No more spare server can be added.
NET_SDK_ERR_ADDED_RD_IS_WD	807	The added spare server is a working server.
NET_SDK_ERR_ADD_ORDER_WRONG	808	Adding flow error.
NET_SDK_ERR_WD_ADD_WD	809	Working server cannot be added to working server.
NET_SDK_ERR_WD_SERVICE_EXCETP	810	CVR service exception (For N+1 mode, it refers to CVR working server exception).
NET_SDK_ERR_RD_SERVICE_EXCETP	811	Spare CVR server exception.
NET_SDK_ERR_ADDED_WD_IS_RD	812	The added working server is spare server.
NET_SDK_ERR_PERFORMANCE_LIMIT	813	The performance reaches the upper-limit.
NET_SDK_ERR_ADDED_DEVICE_EXIST	814	This device already exists.
NET_SDK_ERR_INQUEST_RESUMING	815	Inquest resuming.
NET_SDK_ERR_RECORD_BACKUPING	816	Inquest video backing up.

Error Name	Error Code	Error Description
NET_SDK_ERR_DISK_PLAYING	817	Playing.
NET_SDK_ERR_INQUEST_STARTED	818	Inquest started.
NET_SDK_ERR_LOCAL_OPERATING	819	Locally operating.
NET_SDK_ERR_INQUEST_NOT_START	820	Inquest is not started.
NET_SDK_ERR_CHAN_AUDIO_BIND	821	The channel is not bound or binding two-way audio failed.
NET_DVR_N_PLUS_ONE_MODE	822	Device is in N+1 mode. Cloud storage is not supported.
NET_DVR_CLOUD_STORAGE_OPENED	823	Cloud storage mode is enabled.
NET_DVR_ERR_OPER_NOT_ALLOWED	824	Operation failed. The device is in N+0 taken over status.
NET_DVR_ERR_NEED_RELOCATE	825	The device is in N+0 taken over status. Get re-positioning information and try again.
NET_SDK_ERR_IR_PORT_ERROR	830	IR output error.
NET_SDK_ERR_IR_CMD_ERROR	831	IR output port command number error
NET_SDK_ERR_NOT_INQUESTING	832	Device is not in inquest status.
NET_SDK_ERR_INQUEST_NOT_PAUSED	833	Device is not in paused status.
NET_DVR_CHECK_PASSWORD_MISTAKE_ERROR	834	Incorrect verification code.
NET_DVR_CHECK_PASSWORD_NULL_ERROR	835	Verification code is required.
NET_DVR_UNABLE_CALIB_ERROR	836	Failed to calibrate.
NET_DVR_PLEASE_CALIB_ERROR	837	Calibration first.
NET_DVR_ERR_PANORAMIC_CAL_EMPTY	838	Panoramic calibration is empty in Flash.
NET_DVR_ERR_CALIB_FAIL_PLEASEAGAIN	839	Calibration failed, please try again.

Error Name	Error Code	Error Description
NET_DVR_ERR_DETECTION_LINE	840	Rule line configuration error. Please try again and make sure the line is within the red region.
NET_DVR_EXCEED_FACE_IMAGES_ERROR	843	No more face pictures can be added.
NET_DVR_ANALYSIS_FACE_IMAGES_ERROR	844	Picture recognition failed.
NET_ERR_ALARM_INPUT_OCCUPIED	845	A<-1 alarm number is used for triggering vehicle capture.
NET_DVR_FACELIB_DATABASE_ERROR	846	Database version in face picture library mismatched.
NET_DVR_FACELIB_DATA_ERROR	847	Face picture library data error.
NET_DVR_FACE_DATA_ID_ERROR	848	Invalid face data PID.
NET_DVR_FACELIB_ID_ERROR	849	Invalid face picture library ID.
NET_DVR_EXCEED_FACE_LIBARY_ERROR	850	No more face picture libraries can be established..
NET_DVR_PIC_ANALYSIS_NO_TARGET_ERROR	851	No target recognized in the picture.
NET_DVR_SUBPIC_ANALYSIS_MODELING_ERROR	852	Sub picture modeling failed.
NET_DVR_PIC_ANALYSIS_NO_RESOURCE_ERROR	853	No VCA engine supports picture secondary recognition.
NET_DVR_ANALYSIS_ENGINES_NO_RESOURCE_ERROR	854	No VCA engine.
NET_DVR_ANALYSIS_ENGINES_USAGE_EXCEED_ERROR	855	Overload. The engine CPU reached 100%.
NET_DVR_EXCEED_HUMANMISINFO_FILTER_ENABLED_ERROR	856	No more false alarm channel can be enabled.
NET_DVR_NAME_ERROR	857	Name error.
NET_DVR_NAME_EXIST_ERROR	858	The name already exists.
NET_DVR_FACELIB_PIC_IMPORTING_ERROR	859	The pictures is importing to face picture library.

Error Name	Error Code	Error Description
NET_DVR_PIC_FORMAT_ERROR	864	Invalid picture format.
NET_DVR_PIC_RESOLUTION_INVALID_ERROR	865	Invalid picture resolution.
NET_DVR_PIC_SIZE_EXCEED_ERROR	866	The picture size is too large.
NET_DVR_PIC_ANALYSIS_TARGRT_NUM_EXCEED_ERROR	867	Too many targets in the picture.
NET_DVR_ANALYSIS_ENGINES_LOADING_ERROR	868	Initializing analysis engine.
NET_DVR_ANALYSIS_ENGINES_ABNORMA_ERROR	869	Analysis engine exception.
NET_DVR_ANALYSIS_ENGINES_FACELIB_IMPORTING	870	Analysis engine is importing pictures to face picture library.
NET_DVR_NO_DATA_FOR_MODELING_ERROR	871	No data for modeling.
NET_DVR_FACE_DATA_MODELING_ERROR	872	Device is modeling picture. Concurrent processing is not supported.
NET_ERR_FACELIBDATA_OVERLIMIT	873	No more face picture can be added to the device (the data of imported face picture library)
NET_DVR_ANALYSIS_ENGINES_ASSOCIATED_CHANNEL	874	Channel is linked to the analysis engine.
NET_DVR_ERR_CUSTOMID_LEN	875	The minimum length of upper layer custom ID is 32 bytes.
NET_DVR_ERR_CUSTOMFACELIBID_REPEAT	876	The applied custom face picture library ID is duplicated
NET_DVR_ERR_CUSTOMHUMANID_REPEAT	877	The applied custom person ID is duplicated.
NET_DVR_ERR_URL_DOWNLOAD_FAIL	878	URL download failed.
NET_DVR_ERR_URL_DOWNLOAD_NOTSTART	879	URL download has not started.

Error Name	Error Code	Error Description
NET_DVR_CFG_FILE_SECRETKEY_ERROR	880	The security verification key of configuration file is error.
NET_DVR_THERMOMETRY_REGION_OVERSTEP_ERROR	883	Invalid thermometry region
NET_DVR_ERR_TOO_SHORT_CALIBRATING_TIME	894	Too short time for calibration.
NET_DVR_ERR_AUTO_CALIBRATE_FAILED	895	Auto calibration failed.
NET_DVR_ERR_VERIFICATION_FAILED	896	Verification failed.
NET_DVR_NO_TEMP_SENSOR_ERROR	897	No temperature sensor.
NET_DVR_PUPIL_DISTANCE_OVERSIZE_ERROR	898	The pupil distance is too large.
NET_ERR_WINCHAN_IDX	901	Window channel index error.
NET_ERR_WIN_LAYER	902	Window layer number error(the count of window layers on a single screen exceeds the max number).
NET_ERR_WIN_BLK_NUM	903	Window block number error(the count of screens that single window overlays exceeds the max number).
NET_ERR_OUTPUT_RESOLUTION	904	The output resolution error.
NET_ERR_LAYOUT	905	Layout index error.
NET_ERR_INPUT_RESOLUTION	906	The input resolution is not supported.
NET_ERR_SUBDEVICE_OFFLINE	907	The sub-device is off-line.
NET_ERR_NO_DECODE_CHAN	908	There is no free decoding channel.
NET_ERR_MAX_WINDOW_ABILITY	909	The upper limit of window number.
NET_ERR_ORDER_ERROR	910	Calling order error.
NET_ERR_PLAYING_PLAN	911	Be playing plan.
NET_ERR_DECODER_USED	912	Decoder board is being used.
NET_ERR_OUTPUT_BOARD_DATA_OVERFLOW	913	Output board data overflow
NET_ERR_SAME_USER_NAME	914	Duplicate user name

Error Name	Error Code	Error Description
NET_ERR_INVALID_USER_NAME	915	Invalid user name
NET_ERR_MATRIX_USING	916	Input matrix is in use.
NET_ERR_DIFFERENT_CHAN_TYPE	917	Different channel type (the type of matrix output channel mismatches that of the controller input channel)
NET_ERR_INPUT_CHAN_BINDED	918	Input channel has been bound by other matrix
NET_ERR_BINDED_OUTPUT_CHAN_OVERFLOW	919	The matrix output channels in use exceeded the number bound by matrix and controller
NET_ERR_MAX_SIGNAL_NUM	920	Number of input signals reached upper limit
NET_ERR_INPUT_CHAN_USING	921	Input channel is in use
NET_ERR_MANAGER_LOGON	922	Administrator has logged in, operation failed
NET_ERR_USERALREADY_LOGON	923	The user has logged in, operation failed
NET_ERR_LAYOUT_INIT	924	Scene is initializing, operation failed
NET_ERR_BASEMAP_SIZE_NOT_MATCH	925	Base image size does not match
NET_ERR_WINDOW_OPERATING	926	Window is in other operation, operation failed
NET_ERR_SIGNAL_UPLIMIT	927	Number of signal source window reached upper limit
NET_ERR_WINDOW_SIZE_OVERLIMIT	943	The window size exceeds the limit.
NET_ERR_MAX_WIN_OVERLAP	951	The number of windows overlap has reached the maximum limit.
NET_ERR_STREAMID_CHAN_BOTH_VALID	952	stream ID and channel number are both valid.
NET_ERR_NO_ZERO_CHAN	953	The device has no zero channel.
NEED_RECONNECT	955	Need redirection (for transcoding system)

Error Name	Error Code	Error Description
NET_ERR_NO_STREAM_ID	956	The stream ID does not exist.
NET_DVR_TRANS_NOT_START	957	The transcoding has not been started.
NET_ERR_MAXNUM_STREAM_ID	958	The number of stream ID has reached the maximum limit.
NET_ERR_WORKMODE_MISMATCH	959	The work mode does not match with the requirement.
NET_ERR_MODE_IS_USING	960	It Has been working in current mode.
NET_ERR_DEV_PROGRESSING	961	The device is in processing
NET_ERR_PASSIVE_TRANSCODING	962	It is in transcoding.
NET_DVR_ERR_WINDOW_SIZE_PLACE	975	Wrong window position.
NET_DVR_ERR_RGIONAL_RESTRICTIONS	976	Screen distance exceeds the limit.
NET_DVR_ERR_CLOSE_WINDOWS	984	Operation failed. Close the window first.
NET_DVR_ERR_MATRIX_LOOP_ABILITY	985	Beyond the cycle decoding capacity.
NET_DVR_ERR_MATRIX_LOOP_TIME	986	Invalid cycle decoding time.
NET_DVR_ERR_LINKED_OUT_ABILITY	987	No more linked camera can be added.
NET_ERR_RESOLUTION_NOT_SUPPORT_ODD_VOUT	990	The resolution is not supported (odd No.).
NET_ERR_RESOLUTION_NOT_SUPPORT_EVEN_VOUT	991	The resolution is not supported (even No.).
NET_ERR_UnitConfig_Failed	998	Unit configuration failed.
XML_ABILITY_NOTSUPPORT	1000	Getting capability node is not supported
XML_ANALYZE_NOENOUGH_BUF	1001	Not enough output memory
XML_ANALYZE_FIND_LOCALXML_ERROR	1002	Failed to find related local xml
XML_ANALYZE_LOAD_LOCALXML_ERROR	1003	Loading local xml error

Error Name	Error Code	Error Description
XML_NANLYZE_DVR_DATA_FORMAT_ERROR	1004	Device capability data format error
XML_ANALYZE_TYPE_ERROR	1005	Capability set type error
XML_ANALYZE_XML_NODE_ERROR	1006	XML capability node format error
XML_INPUT_PARAM_ERROR	1007	Input capability XML node value error
XML_VERSION_MISMATCH	1008	XML version does not match
NET_ERR_TRANS_CHAN_START	1101	Transparent channel has been open, operation failed
NET_ERR_DEV_UPGRADING	1102	Device is upgrading
NET_ERR_MISMATCH_UPGRADE_PACK_TYPE	1103	Upgrade pack type does not match
NET_ERR_DEV_FORMATTING	1104	Device is formatting
NET_ERR_MISMATCH_UPGRADE_PACK_VERSION	1105	Upgrade pack version does not match
NET_ERR_PT_LOCKED	1106	PT is locked.
NET_DVR_ERR_ILLEGAL_VERIFICATION_CODE	1111	Illegal verification code. Change the verification code.
NET_DVR_ERR_LACK_VERIFICATION_CODE	1112	No verification code. Enter the verification code.
NET_DVR_ERR_FORBIDDEN_IP	1113	The IP address cannot be configured.
NET_DVR_ERR_HTTP_BKN_EXCEED_ONE	1125	Up to one channel's ANR function can be enabled.
NET_DVR_ERR_FORMATTING_FAILED	1131	Formatting HDD failed.
NET_DVR_ERR_ENCRYPTED_FORMATTING_FAILED	1132	Formatting encrypted HDD failed.
NET_DVR_ERR_WRONG_PASSWORD	1133	Verifying password of SD card failed. Incorrect password.
NET_ERR_SEARCHING_MODULE	1201	Searching peripherals.
NET_ERR_REGISTERING_MODULE	1202	Registering external module
NET_ERR_GETTING_ZONES	1203	Getting arming region parameter
NET_ERR_GETTING_TRIGGER	1204	Getting trigger

Error Name	Error Code	Error Description
NET_ERR_ARMED_STATUS	1205	System is in arming status
NET_ERR_PROGRAM_MODE_STATUS	1206	System is in programming mode
NET_ERR_WALK_TEST_MODE_STATUS	1207	System is in pacing measuring mode
NET_ERR_BYPASS_STATUS	1208	Bypass status
NET_ERR_DISABLED_MODULE_STATUS	1209	Function not enabled
NET_ERR_NOT_SUPPORT_OPERATE_ZONE	1210	Operation is not supported by arming region
NET_ERR_NOT_SUPPORT_MOD_MODULE_ADDR	1211	Module address cannot be modified
NET_ERR_UNREGISTERED_MODULE	1212	Module is not registered
NET_ERR_PUBLIC_SUBSYSTEM_ASSOCIATE_SELF	1213	Public sub system associate with its self
NET_ERR_EXCEEDS_ASSOCIATE_SUBSYSTEM_NUM	1214	Number of associated public sub system reached upper limit
NET_ERR_BE_ASSOCIATED_BY_PUBLIC_SUBSYSTEM	1215	Sub system is associated by other public sub system
NET_ERR_ZONE_FAULT_STATUS	1216	Arming region is in failure status
NET_ERR_SAME_EVENT_TYPE	1217	Same event type exists in enable event trigger alarm output and disable event trigger alarm output
NET_ERR_ZONE_ALARM_STATUS	1218	Arming region is in alarm status
NET_ERR_EXPANSION_BUS_SHORT_CIRCUIT	1219	Extension bus short-circuit
NET_ERR_PWD_CONFLICT	1220	Password conflict, e.g., lock password is identical with duress password
NET_ERR_DETECTOR_GISTERED_BY_OTHER_ZONE	1221	Detector has been registered by other arming regions
NET_ERR_DETECTOR_GISTERED_BY_OTHER_PU	1222	Detector has been registered by other hosts
NET_ERR_DETECTOR_DISCONNECT	1223	Detector offline
NET_ERR_CALL_BUSY	1224	Device in call

Error Name	Error Code	Error Description
NET_ERR_FILE_NAME	1357	File name error, empty or invalid
NET_ERR_BROADCAST_BUSY	1358	Device in broadcast
NET_DVR_ERR_LANENUM_EXCEED	1400	Over the number of lanes.
NET_DVR_ERR_PRAREA_EXCEED	1401	Recognition area is too large.
NET_DVR_ERR_LIGHT_PARAM	1402	Signal lamp access parameters error.
NET_DVR_ERR_LANE_LINE_INVALID	1403	Lane configuration error.
NET_DVR_ERR_STOP_LINE_INVALID	1404	Stop line configuration error.
NET_DVR_ERR_LEFTORRIGHT_LINE_INVALID	1405	Turn left / right boundary configuration error.
NET_DVR_ERR_LANE_NO_REPEAT	1406	Overlay lane number repetition.
NET_DVR_ERR_PRAREA_INVALID	1407	The polygon does not meet the requirements.
NET_DVR_ERR_LIGHT_NUM_EXCEED	1408	Video detection of traffic light signal exceeds the maximum number of.
NET_DVR_ERR_SUBLIGHT_NUM_INVALID	1409	Video detection of traffic signal lamp lights are not legitimate
NET_DVR_ERR_LIGHT_AREASIZE_INVALID	1410	The size of the video detection of traffic light input signal lamp is not valid.
NET_DVR_ERR_LIGHT_COLOR_INVALID	1411	The color of the video detection of traffic light input signal lamp color is not legitimate.
NET_DVR_ERR_LIGHT_DIRECTION_INVALID	1412	The direction property of the video detection of traffic light input light is not valid.
NET_DVR_ERR_LACK_IOABLITY	1413	Lack of IO ability.
NET_DVR_ERR_FTP_PORT	1414	FTP port error.
NET_DVR_ERR_FTP_CATALOGUE	1415	FTP catalogue error.
NET_DVR_ERR_FTP_UPLOAD_TYPE	1416	FTP upload type error.
NET_DVR_ERR_FLASH_PARAM_WRITE	1417	Setting param flash write error.

Error Name	Error Code	Error Description
NET_DVR_ERR_FLASH_PARAM_READ	1418	Getting param flash read error.
NET_DVR_ERR_PICNAME_DELIMITER	1419	Pic name delimiter error.
NET_DVR_ERR_PICNAME_ITEM	1420	Pic name item error.
NET_DVR_ERR_PLATE_RECOGNIZE_TYPE	1421	Plate recognize type error.
NET_DVR_ERR_CAPTURE_TIMES	1422	Capture times error.
NET_DVR_ERR_LOOP_DISTANCE	1423	Loop distance error.
NET_DVR_ERR_LOOP_INPUT_STATUS	1424	Loop input status error.
NET_DVR_ERR_RELATE_IO_CONFLICT	1425	Related IO conflict.
NET_DVR_ERR_INTERVAL_TIME	1426	Interval time error.
NET_DVR_ERR_SIGN_SPEED	1427	Sign speed error.
NET_DVR_ERR_PIC_FLIP	1428	Flip is used.
NET_DVR_ERR_RELATE_LANE_NUMBER	1429	Related lane number error.
NET_DVR_ERR_TRIGGER_MODE	1430	Trigger mode error.
NET_DVR_ERR_DELAY_TIME	1431	Delay time error.
NET_DVR_ERR_EXCEED_RS485_COUNT	1432	Exceed RS485 count.
NET_DVR_ERR_RADAR_TYPE	1433	Radar type error.
NET_DVR_ERR_RADAR_ANGLE	1434	Radar angle error.
NET_DVR_ERR_RADAR_SPEED_VALID_TIME	1435	Radar speed valid time error.
NET_DVR_ERR_RADAR_LINE_CORRECT	1436	Radar line correct error.
NET_DVR_ERR_RADAR_CONST_CORRECT	1437	Radar const correct error.
NET_DVR_ERR_RECORD_PARAM	1438	Record param error.
NET_DVR_ERR_LIGHT_WITHOUT_COLOR_AND_DIRECTION	1439	Light number and other param error.

Error Name	Error Code	Error Description
NET_DVR_ERR_LIGHT_WITHOUT_DETECTION_REGION	1440	Light number and detection region error.
NET_DVR_ERR_RECOGNIZE_PROVINCE_PARAM	1441	Plate recognize Province param error.
NET_DVR_ERR_SPEED_TIMEOUT	1442	IO Speed TimeOut Param error.
NET_DVR_ERR_NTP_TIMEZONE	1443	NTP TimeZone Param error.
NET_DVR_ERR_NTP_INTERVAL_TIME	1444	NTP Interval Time error.
NET_DVR_ERR_NETWORK_CARD_NUM	1445	Network Card Num error.
NET_DVR_ERR_DEFAULT_ROUTE	1446	Default Route error.
NET_DVR_ERR_BONDING_WORK_MODE	1447	Banding Work Mode error.
NET_DVR_ERR_SLAVE_CARD	1448	Sub-Card error.
NET_DVR_ERR_PRIMARY_CARD	1449	Primary Card error.
NET_DVR_ERR_DHCP_PPOE_WORK	1450	DHCP and PPOE not Meanwhile start.
NET_DVR_ERR_NET_INTERFACE	1451	Net Interface invalid.
NET_DVR_ERR_MTU	1452	Invalid MTU parameters.
NET_DVR_ERR_NETMASK	1453	Netmask address invalid.
NET_DVR_ERR_IP_INVALID	1454	IP address invalid.
NET_DVR_ERR_MULTICAST_IP_INVALID	1455	Multicast IP address invalid.
NET_DVR_ERR_GATEWAY_INVALID	1456	Gateway address invalid.
NET_DVR_ERR_DNS_INVALID	1457	DNS Param invalid.
NET_DVR_ERR_ALARMHOST_IP_INVALID	1458	AlarmHost IP invalid.
NET_DVR_ERR_IP_CONFLICT	1459	IP address Conflict.
NET_DVR_ERR_NETWORK_SEGMENT	1460	IP not support Multi Network segment.
NET_DVR_ERR_NETPORT	1461	NetPort error.
NET_DVR_ERR_PPPOE_NOSUPPORT	1462	PPPoE is not supported.

Error Name	Error Code	Error Description
NET_DVR_ERR_DOMAINNAME_NOSUPPORT	1463	Not Support Domain Name.
NET_DVR_ERR_NO_SPEED	1464	Speed Not Enabled.
NET_DVR_ERR_IOSTATUS_INVALID	1465	IO Status invalid.
NET_DVR_ERR_BURST_INTERVAL_INVALID	1466	Burst Interval invalid.
NET_DVR_ERR_RESERVE_MODE	1467	Reserve Mode invalid.
NET_DVR_ERR_LANE_NO	1468	Lane No error.
NET_DVR_ERR_COIL_AREA_TYPE	1469	Coil Area Type error.
NET_DVR_ERR_TRIGGER_AREA_PARAM	1470	Trigger Area Param error.
NET_DVR_ERR_SPEED_LIMIT_PARAM	1471	Speed Limit Param error.
NET_DVR_ERR_LANE_PROTOCOL_TYPE	1472	Lane Protocol Type error.
NET_DVR_ERR_INTERVAL_TYPE	1473	Capture Interval Type error.
NET_DVR_ERR_INTERVAL_DISTANCE	1474	Capture Interval Distance error.
NET_DVR_ERR_RS485_ASSOCIATE_DEVTYPE	1475	Rs485 Associate DevType error.
NET_DVR_ERR_RS485_ASSOCIATE_LANENO	1476	Rs485 Associate LaneNo error.
NET_DVR_ERR_LANENO_ASSOCIATE_MULTIRS485	1477	LaneNo Associate MultRs485 error.
NET_DVR_ERR_LIGHT_DETECTION_REGION	1478	Light Detection Region error.
NET_DVR_ERR_DN2D_NOSUPPORT	1479	UnSupport Capture Frame 2D Noise Reduction.
NET_DVR_ERR_IRISMODE_NOSUPPORT	1480	UnSupport scene Mode.
NET_DVR_ERR_WB_NOSUPPORT	1481	UnSupport White Balance Mode.
NET_DVR_ERR_IO_EFFECTIVENESS	1482	IO Effectiveness invalid.

Error Name	Error Code	Error Description
NET_DVR_ERR_LIGHTNO_MAX	1483	Access Detector Lights Red / Yellow Overrun.
NET_DVR_ERR_LIGHTNO_CONFLICT	1484	Access Detector Lights Red / Yellow Conflict.
NET_DVR_ERR_CANCEL_LINE	1485	Trigger straight line error.
NET_DVR_ERR_STOP_LINE	1486	Subject line area stop line error.
NET_DVR_ERR_RUSH_REDLIGHT_LINE	1487	Red light trigger lines error.
NET_DVR_ERR_IOOUTNO_MAX	1488	IO out port error.
NET_DVR_ERR_IOOUTNO_AHEADTIME_MAX	1489	IO out ahead time error.
NET_DVR_ERR_IOOUTNO_IOWORKTIME	1490	IO out inwork time error.
NET_DVR_ERR_IOOUTNO_FREQMULTI	1491	IO out frequency multiplication error.
NET_DVR_ERR_IOOUTNO_DUTYRATE	1492	IO out duty rate error.
NET_DVR_ERR_VIDEO_WITH_EXPOSURE	1493	IO out work mode error.
NET_DVR_ERR_PLATE_BRIGHTNESS_WITHOUT_FLASHDET	1494	Plate enable in plate compensate mode on.
NET_DVR_ERR_RECOGNIZE_TYPE_PARAM	1495	Recognize Type error.
NET_DVR_ERR_PALTE_RECOGNIZE_AREA_PARAM	1496	Plate Recognize Area Param error.
NET_DVR_ERR_PORT_CONFLICT	1497	Port Conflict.
NET_DVR_ERR_LOOP_IP	1498	IP cannot be the loopback address.
NET_DVR_ERR_DRIVELINE_SENSITIVE	1499	Driveline sensitivity error.
NET_ERR_VQD_TIME_CONFLICT	1500	The time period conflict.
NET_ERR_VQD_PLAN_NO_EXIST	1501	The diagnostic plan of VQD does not exist.
NET_ERR_VQD_CHAN_NO_EXIST	1502	The channel does not exist.

Error Name	Error Code	Error Description
NET_ERR_VQD_CHAN_MAX	1503	The total number of VQD plans exceeds the max limit.
NET_ERR_VQD_TASK_MAX	1504	The total number of VQD tasks exceeds the max limit.
NET_DVR_ERR_EXCEED_MAX_CAPTURE_TIMES	1600	Capture times exceed 2 in flash mode.
NET_DVR_ERR_REDAR_TYPE_CONFLICT	1601	Radar type conflict.
NET_DVR_ERR_LICENSE_PLATE_NULL	1602	The license plate is null.
NET_DVR_ERR_WRITE_DATABASE	1603	Failed to write data into the database.
NET_DVR_ERR_LICENSE_EFFECTIVE_TIME	1604	The effective time of license plate error.
NET_DVR_ERR_PRERECORDED_STARTTIME_LONG	1605	The pre recorded start time is greater than the number of illegal capture.
NET_DVR_ERR_TRIGGER_RULE_LINE	1606	Trigger rule line error.
NET_DVR_ERR_LEFTRIGHT_TRIGGERLINE_NOTVERTICAL	1607	Left and right trigger line is not vertical.
NET_DVR_ERR_FLASH_LAMP_MODE	1608	Flash lamp mode error.
NET_DVR_ERR_ILLEGAL_SNAPSHOT_NUM	1609	Illegal capture number error.
NET_DVR_ERR_ILLEGAL_DETECTION_TYPE	1610	Illegal detection type error.
NET_DVR_ERR_POSITIVEBACK_TRIGGERLINE_HIGH	1611	Positive back to trigger line height error.
NET_DVR_ERR_MIXEDMODE_CAPTYPE_ALLTARGETS	1612	Mixed mode only supports capture type all targets.
NET_DVR_ERR_CARSIGNSPEED_GREATERTHAN_LIMITSPEED	1613	Car sign speed greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_LIMITSPEED	1614	Big car sign speed limit greater than speed limit value.
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_CARSIGNSPEED	1615	Big car sign speed limit is greater than the car sign speed limit value.

Error Name	Error Code	Error Description
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_CARLIMITSPEED	1616	Big car speed limit value is greater than the car speed limit value.
NET_DVR_ERR_BIGCARLOWSPEEDLIMIT_GREATERTHAN_CARLOWSPEEDLIMIT	1617	Big car low speed limit value is greater than the car low speed limit value.
NET_DVR_ERR_CARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1618	Car speed limit greater than exception high speed value.
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1619	Big car speed limit greater than exception high speed value.
NET_DVR_ERR_STOPLINE_MORETHAN_TRIGGERLINE	1620	Stopping more than straight lines trigger lines.
NET_ERR_TIME_OVERLAP	1900	Time periods overlap
NET_ERR_HOLIDAY_PLAN_OVERLAP	1901	Holiday plan overlap
NET_ERR_CARDNO_NOT_SORT	1902	Card number is not sorted
NET_ERR_CARDNO_NOT_EXIST	1903	Card number does not exist
NET_ERR_ILLEGAL_CARDNO	1904	Card number error
NET_ERR_ZONE_ALARM	1905	Arming region is in arming status (parameter cannot be modified)
NET_ERR_ZONE_OPERATION_NOT_SUPPORT	1906	Arming region does not support the operation
NET_ERR_INTERLOCK_ANTI_CONFLICT	1907	Interlock and anti-passback configuration conflict
NET_ERR_DEVICE_CARD_FULL	1908	Card full (return after card reached 10,000)
NET_ERR_HOLIDAY_GROUP_DOWNLOAD	1909	Failed to download holiday group
NET_ERR_LOCAL_CONTROL_OFF	1910	Distributed access controller offline
NET_ERR_LOCAL_CONTROL_DISADD	1911	Distributed access controller is not added
NET_ERR_LOCAL_CONTROL_HASADD	1912	Distributed access controller is added
NET_ERR_LOCAL_CONTROL_DOORNO_CONFLICT	1913	Conflict with added distributed access controller

Error Name	Error Code	Error Description
NET_ERR_LOCAL_CONTROL_COMMUNICATION_FAIL	1914	Distributed access controller communication failed
NET_ERR_OPERAND_INEXISTENCE	1915	Operation object does not exist (operation to door, alarm output, alarm input, return when the object is not added)
NET_ERR_LOCAL_CONTROL_OVER_LIMIT	1916	Distributed access controller exceeded device capability upper limit
NET_ERR_DOOR_OVER_LIMIT	1917	Door exceeded device capability upper limit
NET_ERR_ALARM_OVER_LIMIT	1918	Alarm input and output exceeded device capability upper limit
NET_ERR_LOCAL_CONTROL_ADDRESS_INCONFORMITY_TYPE	1919	Distributed access controller address does not match with type
NET_ERR_NOT_SUPPORT_ONE_MORE_CARD	1920	not support one person multi-card
NET_ERR_DELETE_NO_EXISTENCE_FACE	1921	The face picture does not exist.
NET_ERR_DOOR_SPECIAL_PASSWORD_REPEAT	1922	Repeated door door duress code, the super password, or the dismiss code.
NET_ERR_AUTH_CODE_REPEAT	1923	Repeated device authentication code
NET_ERR_DEPLOY_EXCEED_MAX	1924	No more devices can be armed.
NET_ERR_NOT_SUPPORT_DEL_FP_BY_ID	1925	The fingerprint module does not support deleting fingerprint by finger ID.
NET_ERR_TIME_RANGE	1926	Invalid range of the effective period.
NET_ERR_CAPTURE_TIMEOUT	1927	Collection timed out.
NET_ERR_LOW_SCORE	1928	Low quality of collected data.
NET_ERR_OFFLINE_CAPTURING	1929	The device is collecting data offline and cannot respond.
NET_DVR_ERR_OUTDOOR_COMMUNICATION	1950	Communication exception with outdoor terminal

Error Name	Error Code	Error Description
NET_DVR_ERR_ROOMNO_UNDEFINED	1951	Room number is not set
NET_DVR_ERR_NO_CALLING	1952	No call
NET_DVR_ERR_RINGING	1953	Ringing
NET_DVR_ERR_IS_CALLING_NOW	1954	Call in progress
NET_DVR_ERR_LOCK_PASSWORD_WRONG	1955	Incorrect smart lock password
NET_DVR_ERR_CONTROL_LOCK_FAILURE	1956	Lock control failure
NET_DVR_ERR_CONTROL_LOCK_OVERTIME	1957	Lock control timed out
NET_DVR_ERR_LOCK_DEVICE_BUSY	1958	Smart lock device busy
NET_DVR_ERR_UNOPEN_REMOTE_LOCK_FUNCTION	1959	Remote lock control not enabled
NET_DVR_ERR_FILE_NOT_COMPLETE	2100	Downloaded file is incomplete
NET_DVR_ERR_IPC_EXIST	2101	The camera already exists
NET_DVR_ERR_ADD_IPC	2102	Camera has been added to the channel
NET_DVR_ERR_OUT_OF_RES	2103	Not enough network bandwidth
NET_DVR_ERR_CONFLICT_TO_LOCALIP	2104	IP address of camera conflicts with that of DVR
NET_DVR_ERR_IP_SET	2105	Invalid IP address
NET_DVR_ERR_PORT_SET	2106	Invalid port number
NET_ERR_WAN_NOTSUPPORT	2107	Not in the same LAN, cannot set security question or export GUID file
NET_ERR_MUTEX_FUNCTION	2108	Mutually exclusive function
NET_ERR_QUESTION_CONFIGNUM	2109	Error in number of security question configurations
NET_ERR_FACECHAN_NORESOURCE	2110	All the face VCA channels are occupied.
NET_ERR_DATA_CALLBACK	2111	Data is calling back.

Error Name	Error Code	Error Description
NET_ERR_ATM_VCA_CHAN_IS_RELATED	2112	The VCA channel is already linked.
NET_ERR_ATM_VCA_CHAN_IS_OVERLAPED	2113	The VCA channel is already overlayed.
NET_ERR_FACE_CHAN_UNOVERLAP_EACH_OTHER	2114	The face channels cannot be overlayed.
NET_DVR_SMD_ENCODING_NORESOURCE	2116	Insufficient SMD encoding resource
NET_DVR_SMD_DECODING_NORESOURCE	2117	Insufficient SMD decoding resource
NET_DVR_FACELIB_DATA_PROCESSING	2118	Face picture library data is in processing
NET_DVR_ERR_LARGE_TIME_DIFFRENCE	2119	There is a great time difference between device and server.
NET_DVR_NO_SUPPORT_WITH_PLAYBACK	2120	It is not supported. Playback is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_SMD	2121	It is not supported. SMD of channel is enabled.
NET_DVR_CHANNEL_NO_SUPPORT_WITH_FD	2122	It is not supported. Face capture of channel is enabled.
NET_DVR_ILLEGAL_PHONE_NUMBER	2123	Invalid telephone number
NET_DVR_ILLEGAL_CERITIFICATE_NUMBER	2124	Invalid ID No.
NET_DVR_ERR_CHANNEL_RESOLUTION_NO_SUPPORT	2125	The channel resolution is not supported
NET_DVR_ERR_CHANNEL_COMPRESSION_NO_SUPPORT	2126	The channel encoding format is not supported
NET_DVR_ERR_CLUSTER_DEVICE_TOO_LESS	2127	Deleting is not allowed. The number of devices is not enough
NET_DVR_ERR_CLUSTER_DEL_DEVICE_CM_PAYLOAD	2128	Deleting is not allowed. The device is cluster host.
NET_DVR_ERR_CLUSTER_DEVNUM_OVER_UPPER_LIMIT	2129	No more devices can be added.

Error Name	Error Code	Error Description
NET_DVR_ERR_CLUSTER_DEVICE_TYPE_INCONFORMITY	2130	Device type mismatched.
NET_DVR_ERR_CLUSTER_DEVICE_VERSION_INCONFORMITY	2131	Device version mismatched.
NET_DVR_ERR_CLUSTER_IP_CONFLICT	2132	Cluster system IP address conflict: ipv4 address conflict, invalid ipv6.
NET_DVR_ERR_CLUSTER_IP_INVALID	2133	Invalid cluster system IP address: invalid ipv4, invalid ipv6.
NET_DVR_ERR_CLUSTER_PORT_CONFLICT	2134	Cluster system port conflict
NET_DVR_ERR_CLUSTER_PORT_INVALID	2135	Invalid cluster system port
NET_DVR_ERR_CLUSTER_USERNAEM_OR_PASSWORD_INVALID	2136	Invalid user name or password
NET_DVR_ERR_CLUSTER_DEVICE_ALREADY_EXIST	2137	The device already exists.
NET_DVR_ERR_CLUSTER_DEVICE_NOT_EXIST	2138	The device does not exist.
NET_DVR_ERR_CLUSTER_NON_CLUSTER_MODE	2139	The device working mode is not the cluster mode .
NET_DVR_ERR_CLUSTER_IP_NOT_SAME_LAN	2140	IP addresses are in different LAN. Building cluster or extending capacity for NVRs in different LAN is not allowed.
NET_DVR_ERR_IDENTITY_KEY	2147	Incorrect interaction password
NET_DVR_MISSING_IDENTITY_KEY	2148	Interaction password is missing
NET_DVR_ERR_CAPTURE_PACKAGE_FAILED	2141	Capturing packets failed.
NET_DVR_ERR_CAPTURE_PACKAGE_PROCESSING	2142	Capturing packet.
NET_DVR_ERR_SAFETY_HELMET_NO_RESOURCE	2143	No enough hard hat detection resource.

Error Name	Error Code	Error Description
NET_DVR_NO_SUPPORT_WITH_ABSTRACT	2144	This function is not supported. Video synopsis is already enabled.
NET_DVR_INSUFFICIENT_DEEP_LEARNING_RESOURCES	2146	No more deep learning resources can be added.
NET_DVR_NO_SUPPORT_WITH_PERSON_DENSITY_DETECT	2149	People gathering density is enabled, it is not supported
NET_DVR_IPC_RESOLUTION_OVERFLOW	2150	The network camera resolution is too large
NET_DVR_IPC_BITRATE_OVERFLOW	2151	The network camera bitrate is too large
NET_DVR_ERR_INVALID_TASKID	2152	Invalid taskID
NET_DVR_PANEL_MODE_NOT_CONFIG	2153	The ATM panel mode is not configured.
NET_DVR_NO_HUMAN_ENGINES_RESOURCE	2154	No enough engine resource
NET_DVR_ERR_TASK_NUMBER_OVERFLOW	2155	No more task data is allowed
NET_DVR_ERR_COLLISION_TIME_OVERFLOW	2156	Collision time is over the limit
NET_DVR_ERR_EVENT_NOTSUPPORT	2159	Subscribing alarm/event is not supported.
NET_DVR_IPC_NUM_REACHES_LIMIT	2184	The max. number of network camera channels reached.
NET_DVR_IOT_NUM_REACHES_LIMIT	2185	The max. number of IoT channels reached
NET_DVR_IOT_CHANNEL_DEVICE_EXIST	2186	Device of the IoT channel already exists.
NET_DVR_IOT_CHANNEL_DEVICE_NOT_EXIST	2187	Device of the IoT channel does not exist.
NET_DVR_INVALID_IOT_PROTOCOL_TYPE	2188	Invalid IoT protocol type
NET_DVR_INVALID_EZVIZ_SECRET_KEY	2189	Invalid verification code

Error Name	Error Code	Error Description
NET_DVR_DUPLICATE_IOT_DEVICE	2190	Duplicated IoT device
NET_DVR_ERROR_NEED_DOUBLE_VERIFICATION	2206	Double verification is required
NET_DVR_NO_DOUBLE_VERIFICATION_USER	2207	No double verification user
NET_DVR_TIMESPAN_NUM_OVER_LIMIT	2209	Max. number of time buckets reached
NET_DVR_CHANNEL_NUM_OVER_LIMIT	2210	Max. number of channels reached
NET_DVR_NO_SEARCH_ID_RESOURCE	2211	Insufficient searchID resources
NET_DVR_SWITCH_TIMEDIFF_LESS_LIMIT	2249	Time difference between power on and off should be less than 10 minutes.
NET_DVR_NO_SUPPORT_DELETE_STRANGER_LIB	2262	Deleting stranger library is not supported
NET_DVR_NO_SUPPORT_CREATE_STRANGER_LIB	2263	Creating stranger library is not supported
NET_DVR_SSD_FILE_SYSTEM_ERROR	2266	SSD file system error
NET_DVR_INSUFFICIENT_SSD_FOR_FPD	2267	Insufficient SSD space for person frequency detection
NET_DVR_SMRDISK_NOT_SUPPORT_RAID	2269	SMR disk does not support RAID.
NET_DVR_ERR_NOTSUPPORT_DEICING	3001	Device does not support deicing function under current status.(Deicing function is only supported under the power status of POE+, AC24V, and DC12V).
NET_DVR_ERR_THERMENABLE_CLOSE	3002	Temperature measurement function is not enabled. (The enable function in NET_DVR_THERMOMETRY_BASICPARAM is not turned on)
NET_DVR_ERR_PANORAMIC_LIMIT_OPERATED	3004	Panoramic map and limit cannot be operated at same time

Error Name	Error Code	Error Description
NET_DVR_ERR_SMARTH264_ROI_OPERATED	3005	SmartH264 and ROI cannot be enabled at the same time.
NET_DVR_ERR_RULENUM_LIMIT	3006	No more rules can be added.
NET_DVR_ERR_LASER_DEICING_OPERATED	3007	Laser and deicing function cannot be enabled at the same time.
NET_DVR_ERR_OFFDIGITALZOOM_OR_MINZOOMLIMIT	3008	Please disable the digital zoom function or set the zoom limit to the minimum value. Otherwise, when enabling smoke and fire detection, behavior analysis, ship detection, defective point correction, temperature measurement, smoke and fire shielding function, this error code will be prompted.
NET_DVR_SYNCHRONIZEFOV_ERROR	3010	Field of view synchronization failed.
NET_DVR_RULE_SHIELDMASK_CONFLICT_ERROR	3013	The rule region conflicts with the shielded area.
NET_DVR_ERR_NO_SAFETY_HELMET_REGION	3501	The hard hat detection area is not configured.
NET_DVR_ERR_UNCLOSED_SAFETY_HELMET	3502	The hard hat detection is enabled.
NET_DVR_UPLOAD_HBDLIBID_ERROR	3504	Incorrect ID of human body picture library (incorrect HBDID or customHBDID)

### RTSP Communication Library Related Errors

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NOENOUGHPRI	401	Authentication failed: if server returns 401, it will change to this error code
NET_DVR_RTSP_ERROR_ALLOC_RESOURCE	402	Failed to allocate the resource
NET_DVR_RTSP_ERROR_PARAMETER	403	Parameter error

Error Name	Error Code	Error Description
NET_DVR_RTSP_ERROR_NO_URL	404	The assigned URL does not exist: when the server returns 404, SDK turns to this error code. E.g. the channel is not available, or the channel does not support sub stream
NET_DVR_RTSP_ERROR_FORCE_STOP	406	The user forces to exit midway
NET_DVR_RTSP_GETPORTFAILED	407	RTSP port getting error.
NET_DVR_RTSP_DESCRIBERROR	410	RTSP DECRIBE communicate error
NET_DVR_RTSP_DESCRIBESENDTIMEOUT	411	Sending "RTSP DECRIBE" is timeout.
NET_DVR_RTSP_DESCRIBESENDEROR	412	Failed to send "RTSP DECRIBE".
NET_DVR_RTSP_DESCRIBERCVTIMEOUT	413	Receiving "RTSP DECRIBE" is timeout.
NET_DVR_RTSP_DESCRIBERECDATALOST	414	Receiving data of "RTSP DECRIBE" error.
NET_DVR_RTSP_DESCRIBERECCROR	415	Failed to receive "RTSP DECRIBE".
NET_DVR_RTSP_DESCRIBESERVERERR	416	"RTSP DECRIBE, the device returns the error code: 501 (failed to allocate the resource in the device)
NET_DVR_RTSP_SETUPERROR	420	(or 419), RTSP SETUP interaction error. Generally, it is that the address(URL) returned by the device is not accessible, or it is rejected by the server
NET_DVR_RTSP_SETUPSENDTIMEOUT	421	Sending "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPSENDEROR	422	Sending "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	Receiving "RTSP SETUP" is timeout.
NET_DVR_RTSP_SETUPRECDATALOST	424	Receiving data of "RTSP SETUP" error.
NET_DVR_RTSP_SETUPRECVEROR	425	Failed to receive "RTSP SETUP".
NET_DVR_RTSP_OVER_MAX_CHAN	426	"RTSP SETUP" device returns the error that values 401 or 501. It

Error Name	Error Code	Error Description
		exceeds the max connection number.
NET_DVR_RTSP_PLAYERROR	430	RTSP PLAY interaction error.
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	Sending "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYSENDERRORE	432	Sending "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	Receiving "RTSP PLAY" is timeout.
NET_DVR_RTSP_PLAYRECVDATALOST	434	Receiving data of "RTSP PLAY" error.
NET_DVR_RTSP_PLAYRECVERROR	435	Failed to receive "RTSP PLAY".
NET_DVR_RTSP_PLAYSERVERERR	436	"RTSP PLAY" device returns the error that values 401 or 501.
NET_DVR_RTSP_TEARDOWNERROR	440	RTSP TEARDOWN interaction error.
NET_DVR_RTSP_TEARDOWNSENDFTIMEOUT	441	Sending "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_TEARDOWNSENDERRORE	442	Sending "RTSP TEARDOWN" error.
NET_DVR_RTSP_TEARDOWNRECVTIMEOUT	443	Receiving "RTSP TEARDOWN" is timeout.
NET_DVR_RTSP_TEARDOWNRECVDATALOST	444	Receiving data of "RTSP TEARDOWN" error.
NET_DVR_RTSP_TEARDOWNRECVERROR	445	Failed to receive "RTSP TEARDOWN".
NET_DVR_RTSP_TEARDOWNSERVERERR	446	"RTSP TEARDOWN" device returns the error that values 401 or 501.

## Software Decoding Library Related Errors

Error Name	Error Code	Error Description
NET_PLAYM4_NOERROR	500	No error.
NET_PLAYM4_PARA_OVER	501	Input parameter is invalid.
NET_PLAYM4_ORDER_ERROR	502	API calling order error.
NET_PLAYM4_TIMER_ERROR	503	Failed to create multimedia clock.

Error Name	Error Code	Error Description
NET_PLAYM4_DEC_VIDEO_ERROR	504	Failed to decode video data.
NET_PLAYM4_DEC_AUDIO_ERROR	505	Failed to decode audio data.
NET_PLAYM4_ALLOC_MEMORY_ERROR	506	Failed to allocate memory.
NET_PLAYM4_OPEN_FILE_ERROR	507	Failed to open the file.
NET_PLAYM4_CREATE_OBJ_ERROR	508	Failed to create thread event.
NET_PLAYM4_CREATE_DDRAW_ERROR	509	Failed to create DirectDraw object.
NET_PLAYM4_CREATE_OFSSCREEN_ERROR	510	Failed to create backstage cache for OFFSCREEN mode.
NET_PLAYM4_BUF_OVER	511	Buffer overflow, failed to input stream.
NET_PLAYM4_CREATE_SOUND_ERROR	512	Failed to create audio equipment.
NET_PLAYM4_SET_VOLUME_ERROR	513	Failed to set the volume.
NET_PLAYM4_SUPPORT_FILE_ONLY	514	This API can be called only for file playback mode.
NET_PLAYM4_SUPPORT_STREAM_ONLY	515	This API can be called only when playing stream.
NET_PLAYM4_SYS_NOT_SUPPORT	516	Not support by the system. Decoder can only work on the system above Pentium 3.
NET_PLAYM4_FILEHEADER_UNKNOWN	517	There is no file header.
NET_PLAYM4_VERSION_INCORRECT	518	The version mismatch between decoder and encoder.
NET_PLAYM4_INIT_DECODER_ERROR	519	Failed to initialize the decoder.
NET_PLAYM4_CHECK_FILE_ERROR	520	The file is too short, or the stream data is unknown.
NET_PLAYM4_INIT_TIMER_ERROR	521	Failed to initialize multimedia clock.
NET_PLAYM4_BLT_ERROR	522	BLT failure.

Error Name	Error Code	Error Description
NET_PLAYM4_UPDATE_ERROR	523	Failed to update overlay surface
NET_PLAYM4_OPEN_FILE_ERROR_MULTI	524	Failed to open video & audio stream file.
NET_PLAYM4_OPEN_FILE_ERROR_VIDEO	525	Failed to open video stream file.
NET_PLAYM4_JPEG_COMPRESS_ERROR	526	JPEG compression error.
NET_PLAYM4_EXTRACT_NOT_SUPPORT	527	Don't support the version of this file.
NET_PLAYM4_EXTRACT_DATA_ERROR	528	Extract video data failed.

### Container Format Conversion Library Related Errors

Error Name	Error Code	Error Description
NET_CONVERT_ERROR_NOT_SUPPORT	581	This container format is not supported.

### Two Way Audio Library Related Errors

Error Name	Error Code	Error Description
NET_AUDIOINTERCOM_OK	600	No error.
NET_AUDIOINTECOM_ERR_NOTSUPORT	601	Not support.
NET_AUDIOINTECOM_ERR_ALLOC_MEMORY	602	Memory allocation error.
NET_AUDIOINTECOM_ERR_PARAMETER	603	Parameter error.
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	API calling order error.
NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	No audio device
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	Failed to open the audio device
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	Context error.
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV file error.
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	The type of WAV parameter is invalid

Error Name	Error Code	Error Description
NET_AUDIOINTECOM_ERR_ENCODE_FAIL	610	Failed to encode data
NET_AUDIOINTECOM_ERR_DECODE_FAIL	611	Failed to decode data
NET_AUDIOINTECOM_ERR_NO_PLAYBACK	612	Failed to play audio
NET_AUDIOINTECOM_ERR_DENOISE_FAIL	613	Failed to denoise
NET_AUDIOINTECOM_ERR_UNKOWN	619	Unknown

### QoS Stream Control Library Related Errors

Error Name	Error Code	Error Description
NET_QOS_ERR_SCHEDPARAMS_BAD_MINIMUM_INTERVAL	678	Incorrect predefined minimum interval.
NET_QOS_ERR_SCHEDPARAMS_BAD_FRACTION	679	Incorrect predefined score.
NET_QOS_ERR_SCHEDPARAMS_INVALID_BANDWIDTH	680	Invalid predefined bandwidth.
NET_QOS_ERR_PACKET_TOO_BIG	687	The packet size is too large.
NET_QOS_ERR_PACKET_LENGTH	688	Invalid packet size.
NET_QOS_ERR_PACKET_VERSION	689	Incorrect packet version information.
NET_QOS_ERR_PACKET_UNKNOW	690	Unknown packet.
NET_QOS_ERR_OUTOFMEM	695	Out of memory.
NET_QOS_ERR_LIB_NOT_INITIALIZED	696	The library is not initialized.
NET_QOS_ERR_SESSION_NOT_FOUND	697	No session found.
NET_QOS_ERR_INVALID_ARGUMENTS	698	Invalid parameters.
NET_QOS_ERROR	699	QoS Stream Control Library error.
NET_QOS_OK	700	No error.

## NPQ (Network Protocol Quality) Related Error

Error Name	Error Code	Error Description
NET_ERR_NPQ_PARAM	8001	NPQ library: Incorrect parameter.
NET_ERR_NPQ_SYSTEM	8002	NPQ library: Operating system error.
NET_ERR_NPQ_GENRAL	8003	NPQ library: Internal error.
NET_ERR_NPQ_PRECONDITION	8004	NPQ library: Calling sequence error.
NET_ERR_NPQ_NOTSUPPORT	8005	NPQ library: This function is not supported.
NET_ERR_NPQ_NOTCALLBACK	8100	No data is called back.
NET_ERR_NPQ_LOADLIB	8101	Loading NPQ library failed.
NET_ERR_NPQ_STEAM_CLOSE	8104	The NPQ function of this stream is not enabled.
NET_ERR_NPQ_MAX_LINK	8110	No more streaming channel's NPQ function can be enabled.
NET_ERR_NPQ_STREAM_CFG_CONFLICT	8111	The configured encoding parameters conflicted.

## Appendix E. Request URIs

Description	URI	Method	Request and Response Message
Get device information.	/ISAPI/System/deviceInfo	GET	XML_DeviceInfo XML_ResponseStatus
Edit device information.	/ISAPI/System/deviceInfo	PUT	-
Control PTZ.	/ISAPI/PTZCtrl/channels/<ID>/continuous	PUT	XML_ResponseStatus
Get preset list.	/ISAPI/PTZCtrl/channels/<ID>/presets	GET	XML_PTZPresetList XML_ResponseStatus
Manage all configured presets.	/ISAPI/PTZCtrl/channels/<ID>/presets	POST	-
Delete all presets.	/ISAPI/PTZCtrl/channels/<ID>/presets	DELETE	-
Add a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	PUT	XML_ResponseStatus
Delete a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	DELETE	XML_ResponseStatus
Get a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>	GET	-
Call a preset.	/ISAPI/PTZCtrl/channels/<ID>/presets/<ID>/goto	PUT	XML_ResponseStatus
Get partition status.	/ISAPI/SecurityCP/status/subSystems?format=json	GET	JSON_SubSysList JSON_ResponseStatus
Arm a partition.	/ISAPI/SecurityCP/control/arm/<ID>?ways=<string>&format=json	PUT	JSON_ResponseStatus
Disarm a partition.	/ISAPI/SecurityCP/control/disarm/<ID>?format=json	PUT	JSON_ResponseStatus
Clear partition alarms.	/ISAPI/SecurityCP/control/clearAlarm/<ID>?format=json	PUT	JSON_ResponseStatus
Get zone status	/ISAPI/SecurityCP/status/zones?format=json	GET	JSON_ZoneList JSON_ResponseStatus

Search partition status according to conditions.	/ISAPI/SecurityCP/status/zones?format=json	POST	-
Zone bypass.	/ISAPI/SecurityCP/control/bypass?format=json	PUT	JSON_ResponseStatus
Recover bypass of multiple zones.	/ISAPI/SecurityCP/control/bypassRecover?format=json	PUT	JSON_ResponseStatus
Get relay status by specific conditions.	/ISAPI/SecurityCP/status/outputStatus?format=json	POST	JSON_OutputSearch JSON_ResponseStatus
Control relay in batch.	/ISAPI/SecurityCP/control/outputs?format=json	POST	JSON_ResponseStatus
Get the information of all I/O output ports.	/ISAPI/System/IO/outputs	GET	XML_IOutputPortList XML_ResponseStatus
Get status of a specific alarm output.	/ISAPI/System/IO/outputs/<ID>/status	GET	XML_IOPortStatus XML_ResponseStatus
Manually trigger a specific alarm output.	/ISAPI/System/IO/outputs/<ID>/trigger	PUT	XML_ResponseStatus
Get device time zone.	/ISAPI/System/time	GET	XML_TimeData XML_ResponseStatus
Get or set device time parameters.	/ISAPI/System/time	PUT	-
Operations about management of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels	GET	XML_InputProxyChannelList XML_ResponseStatus
Configure operations about management of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels	PUT	-
Create digital channels	/ISAPI/ContentMgmt/InputProxy/channels	POST	-

Get status of all digital channels.	/ISAPI/ContentMgmt/InputProxy/channels/status	GET	XML_InputProxyChannelStatusList XML_ResponseStatus
Refresh the video mode manually before playback.	/ISAPI/ContentMgmt/record/control/manualRefresh/channels/<ID>	PUT	XML_ResponseStatus
Search for access control events.	/ISAPI/AccessControl/AcsEvent?format=json	POST	JSON_AcsEvent XML_ResponseStatus
Search for person information.	/ISAPI/AccessControl/UserInfo/Search?format=json	POST	JSON_UserInfoSearch XML_ResponseStatus

## E.1 /ISAPI/AccessControl/AcsCfg/capabilities?format=json

Get the configuration capability of the access controller.

### Request URI Definition

**Table B-1 GET /ISAPI/AccessControl/AcsCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the access controller.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_Cap_AcsCfg</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

## E.2 /ISAPI/AccessControl/AcsCfg?format=json

Operations about the configuration of the access controller.

### Request URI Definition

**Table B-2 GET /ISAPI/AccessControl/AcsCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration parameters of the access controller.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_AcsCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-3 PUT /ISAPI/AccessControl/AcsCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the access controller.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_AcsCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

### E.3 /ISAPI/AccessControl/AcsEvent/capabilities?format=json

Get the capability of searching for access control events

#### Request URI Definition

**Table B-4 GET /ISAPI/AccessControl/AcsEvent/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of searching for access control events.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_AcsEvent</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.4 /ISAPI/AccessControl/AcsEvent/StorageCfg/capabilities?format=json

Get the storage configuration capability of access control events.

### Request URI Definition

**Table B-5 GET /ISAPI/AccessControl/AcsEvent/StorageCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the storage configuration capability of access control events.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_EventStorageCfgCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.5 /ISAPI/AccessControl/AcsEvent/StorageCfg?format=json

Get or set the storage parameters of access control events.

### Request URI Definition

**Table B-6 GET /ISAPI/AccessControl/AcsEvent/StorageCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the storage parameters of access control events.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_EventStorageCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-7 PUT /ISAPI/AccessControl/AcsEvent/StorageCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the storage parameters of access control events.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

<b>Request</b>	<u><a href="#">JSON_EventStorageCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.6 /ISAPI/AccessControl/AcsEvent?format=json

Search for access control events.

### Request URI Definition

**Table B-8 POST /ISAPI/AccessControl/AcsEvent?format=json**

<b>Method</b>	POST
<b>Description</b>	Search for access control events.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_AcsEventCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_AcsEvent</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

### Remarks

- The recommended timeout of this URI is 10 seconds.
- If the response message contains picture data, the picture data will be returned by boundary method; otherwise, the response message in JSON format will be returned directly.

### Example

Sample Response Message with Picture Data

```
--MIME_boundary
Content-Type: application/json
Content-Length:480

{
    "AcsEvent": {
        "searchID": "",
        "responseStatusStrg": "OK",
        "numOfMatches": 1,
        "totalMatches": 1,
        "InfoList": [
            {
                "major": 1,
                "minor": 1,
                "time": "2016-12-12T17:30:08+08:00",
                "netUser": "",
                "remoteHostAddr": "",
                "cardNo": ""
            }
        ]
    }
}
```

```
        "cardType":1,
        "whiteListNo":1,
        "reportChannel":1,
        "cardReaderKind":1,
        "cardReaderNo":1,
        "doorNo":1,
        "verifyNo":1,
        "alarmInNo":1,
        "alarmOutNo":1,
        "caseSensorNo":1,
        "RS485No":1,
        "multiCardGroupNo":1,
        "accessChannel":1,
        "deviceNo":1,
        "distractControlNo":1,
        "employeeNoString":"",
        "localControllerID":1,
        "InternetAccess":1,
        "type":1,
        "MACAddr":"",
        "swipeCardType":1,
        "serialNo":1,
        "channelControllerID":1,
        "channelControllerLampID":1,
        "channelControllerIRAdaptorID":1,
        "channelControllerIREmitterID":1,
        "userType":"normal",
        "currentVerifyMode":"",
        "attendanceStatus":"",
        "statusValue":1,
        "pictureURL":"",
        "picturesNumber":1,
        "filename":"picture1"
    ],
}
}
--MIME_boundary
Content-Disposition: form-data; filename="picture1"; //Picture data
Content-Type:image/jpeg
Content-Length:12345

fgagashgshdasdad...
--MIME_boundary--
```

### E.7 /ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json

Get the capability of getting total number of access control events by specific conditions.

## Request URI Definition

**Table B-9 GET /ISAPI/AccessControl/AcsEventTotalNum/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of getting total number of access control events by specific conditions.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_AcsEventTotalNum</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.8 /ISAPI/AccessControl/AcsEventTotalNum?format=json

Get the total number of access control events by specific conditions.

## Request URI Definition

**Table B-10 POST /ISAPI/AccessControl/AcsEventTotalNum?format=json**

<b>Method</b>	POST
<b>Description</b>	Get the total number of access control events by specific conditions.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	<u><a href="#">JSON_AcsEventTotalNumCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_AcsEventTotalNum</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

- The recommended timeout is 30s.
- This URI is not supported by integration of information release system.

## E.9 /ISAPI/AccessControl/AntiSneakCfg/capabilities?format=json

Get the anti-passing back configuration capability.

### Request URI Definition

**Table B-11 GET /ISAPI/AccessControl/AntiSneakCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the anti-passing back configuration capability.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_AntiSneakCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.10 /ISAPI/AccessControl/AntiSneakCfg?format=json

Operations about anti-passing back configuration.

### Request URI Definition

**Table B-12 GET /ISAPI/AccessControl/AntiSneakCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the anti-passing back configuration parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_AntiSneakCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-13 PUT /ISAPI/AccessControl/AntiSneakCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the anti-passing back parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_AntiSneakCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.11 /ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json

Get or set the parameters of the attendance schedule template.

### Request URI Definition

**Table B-14 GET /ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of the attendance schedule template.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_AttendancePlanTemplate</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-15 PUT /ISAPI/AccessControl/Attendance/planTemplate/<TemplateNo>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the attendance schedule template.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_AttendancePlanTemplate</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

### Remarks

The <TemplateNo> in the request URI refers to the attendance schedule template No.

## E.12 /ISAPI/AccessControl/Attendance/planTemplate/capabilities?format=json

Get the configuration capability of the attendance schedule template.

### Request URI Definition

Table B-16 GET /ISAPI/AccessControl/Attendance/planTemplate/capabilities?format=json

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the attendance schedule template.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_AttendancePlanTemplateCap</i> Failed: <i>JSON_ResponseStatus</i>

## E.13 /ISAPI/AccessControl/Attendance/planTemplate?format=json

Get the list of attendance schedule templates.

### Request URI Definition

Table B-17 GET /ISAPI/AccessControl/Attendance/planTemplate?format=json

<b>Method</b>	GET
<b>Description</b>	Get the list of attendance schedule templates.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_AttendancePlanTemplateList</i> Failed: <i>JSON_ResponseStatus</i>

## E.14 /ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json

Get or set the parameters of the week attendance schedule.

## Request URI Definition

**Table B-18 GET /ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of the week attendance schedule.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <b><i>JSON_AttendanceWeekPlan</i></b> Failed: <b><i>JSON_ResponseStatus</i></b>

**Table B-19 PUT /ISAPI/AccessControl/Attendance/weekPlan/<PlanNo>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the week attendance schedule.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<b><i>JSON_AttendanceWeekPlan</i></b>
<b>Response</b>	<b><i>JSON_ResponseStatus</i></b>

## Remarks

The <PlanNo> in the request URI refers to the attendance schedule No.

## E.15 /ISAPI/AccessControl/Attendance/weekPlan/capabilities? format=json

Get the configuration capability of the week attendance schedule.

## Request URI Definition

**Table B-20 GET /ISAPI/AccessControl/Attendance/weekPlan/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the week attendance schedule.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <b><i>JSON_AttendanceWeekPlanCap</i></b>

	Failed: <i>JSONResponseStatus</i>
--	-----------------------------------

## E.16 /ISAPI/AccessControl/blackObject/capabilities?format=json

Get the configuration capability of the black body.

### Request URI Definition

**Table B-21 GET /ISAPI/AccessControl/blackObject/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the black body.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_BlackBodyCfg</i> Failed: <i>JSONResponseStatus</i>

## E.17 /ISAPI/AccessControl/blackObject?format=json

Get or set the black body parameters.

### Request URI Definition

**Table B-22 GET /ISAPI/AccessControl/blackObject?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the black body parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_BlackBodyCfg</i> Failed: <i>JSONResponseStatus</i>

**Table B-23 PUT /ISAPI/AccessControl/blackObject?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the black body parameters.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON BlackBodyCfg</a></u>
<b>Response</b>	<u><a href="#">JSON ResponseStatus</a></u>

## E.18 /ISAPI/AccessControl/capabilities

Get the functional capability of access control.

### Request URI Definition

**Table B-24 GET /ISAPI/AccessControl/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the functional capability of access control.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML Cap_AccessControl</a></u> Failed: <u><a href="#">XML ResponseStatus</a></u>

## E.19 /ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json

Get the capability of collecting card information.

### Request URI Definition

**Table B-25 GET /ISAPI/AccessControl/CaptureCardInfo/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of collecting card information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON CardInfoCap</a></u> Failed: <u><a href="#">JSON ResponseStatus</a></u>

## E.20 /ISAPI/AccessControl/CaptureCardInfo?format=json

Collect card information.

### Request URI Definition

**Table B-26 GET /ISAPI/AccessControl/CaptureCardInfo?format=json**

<b>Method</b>	GET
<b>Description</b>	Collect card information by the card reading module of the device.
<b>Query</b>	<p><b>format:</b> determine the format of request or response message.</p> <p><b>security:</b> the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the value of the field <b>cardNo</b> should be encrypted.</p> <p><b>iv:</b> the initialization vector, and it is required when <b>security</b> is 1 or 2.</p>
<b>Request</b>	None.
<b>Response</b>	<p>Succeeded: <u><a href="#">JSON CardInfo Collection</a></u></p> <p>Failed: <u><a href="#">JSON ResponseStatus</a></u></p>

## E.21 /ISAPI/AccessControl/CaptureFingerPrint

Collect fingerprint information.

### Request URI Definition

**Table B-27 POST /ISAPI/AccessControl/CaptureFingerPrint**

<b>Method</b>	POST
<b>Description</b>	Collect fingerprint information.
<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML CaptureFingerPrintCond</a></u>
<b>Response</b>	<p>Succeeded: <u><a href="#">XML CaptureFingerPrint</a></u></p> <p>Failed: <u><a href="#">XML ResponseStatus</a></u></p>

## E.22 /ISAPI/AccessControl/CaptureFingerPrint/capabilities

Get the fingerprint collection capability.

### Request URI Definition

**Table B-28 GET /ISAPI/AccessControl/CaptureFingerPrint/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the fingerprint collection capability.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_CaptureFingerPrint</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

## E.23 /ISAPI/AccessControl/CaptureIDInfo/capabilities?format=json

Get the capability of collecting ID card information.

### Request URI Definition

**Table B-29 GET /ISAPI/AccessControl/CaptureIDInfo/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of collecting ID card information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_IdentityInfoCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.24 /ISAPI/AccessControl/CaptureIDInfo?format=json

Collect ID card information.

## Request URI Definition

**Table B-30 POST /ISAPI/AccessControl/CaptureIDInfo?format=json**

<b>Method</b>	POST
<b>Description</b>	Collect ID card information.
<b>Query</b>	<b>security</b> : the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the value of the field <b>IDCardNo</b> should be encrypted. <b>iv</b> : the initialization vector, and it is required when <b>security</b> is 1 or 2. <b>format</b> : determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_IdentityInfoCond</a></u>
<b>Response</b>	<u><a href="#">JSON_IdentityInfo</a></u>

## E.25 /ISAPI/AccessControl/CapturePresetParam/capabilities?format=json

Get the configuration capability of online collection preset parameters.

## Request URI Definition

**Table B-31 GET /ISAPI/AccessControl/CapturePresetParam/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of online collection preset parameters.
<b>Query</b>	<b>format</b> : determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CapturePresetCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.26 /ISAPI/AccessControl/CapturePresetParam?format=json

Get or set the online collection preset parameters.

### Request URI Definition

**Table B-32 GET /ISAPI/AccessControl/CapturePresetParam?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the online collection preset parameters.
<b>Query</b>	<p><b>format:</b> determine the format of request or response message.</p> <p><b>security:</b> the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the value of the field <b>name</b> should be encrypted.</p> <p><b>iv:</b> the initialization vector, and it is required when <b>security</b> is 1 or 2.</p>
<b>Request</b>	None.
<b>Response</b>	<p>Succeeded: <u><a href="#">JSON_CapturePreset</a></u></p> <p>Failed: <u><a href="#">JSON_ResponseStatus</a></u></p>

**Table B-33 PUT /ISAPI/AccessControl/CapturePresetParam?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the online collection preset parameters.
<b>Query</b>	<p><b>format:</b> determine the format of request or response message.</p> <p><b>security:</b> the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the value of the field <b>name</b> should be encrypted.</p> <p><b>iv:</b> the initialization vector, and it is required when <b>security</b> is 1 or 2.</p>
<b>Request</b>	<u><a href="#">JSON_CapturePreset</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.27 /ISAPI/AccessControl/CaptureRule/capabilities?format=json

Get the configuration capability of online collection rules.

### Request URI Definition

**Table B-34 GET /ISAPI/AccessControl/CaptureRule/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of online collection rules.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CaptureRuleCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.28 /ISAPI/AccessControl/CaptureRule?format=json

Get or set the parameters of online collection rules.

### Request URI Definition

**Table B-35 GET /ISAPI/AccessControl/CaptureRule?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of online collection rules.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CaptureRule</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-36 PUT /ISAPI/AccessControl/CaptureRule?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of online collection rules.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

Request	<u>JSON_CaptureRule</u>
Response	<u>JSON_ResponseStatus</u>

## E.29 /ISAPI/AccessControl/CardInfo/capabilities?format=json

Get the card management capability.

### Request URI Definition

**Table B-37 GET /ISAPI/AccessControl/CardInfo/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the card management capability.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>JSON_Cap_CardInfo</u> Failed: <u>JSON_ResponseStatus</u>

## E.30 /ISAPI/AccessControl/CardInfo/Count?format=json

Get the total number of the added cards.

### Request URI Definition

**Table B-38 GET /ISAPI/AccessControl/CardInfo/Count?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the total number of the added cards.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management

	server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CardInfoCount</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

This URI is not supported by integration of information release system.

### E.31 /ISAPI/AccessControl/CardInfo/Count? format=json&employeeNo=<ID>

Get the number of cards linked with a specific person.

#### Request URI Definition

Table B-39 GET /ISAPI/AccessControl/CardInfo/Count?format=json&employeeNo=<ID>

<b>Method</b>	GET
<b>Description</b>	Get the number of cards linked with a specific person.
<b>Query</b>	<b>format</b> : determine the format of request or response message. <b>employeeNo</b> : employee No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CardInfoCount</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

The <ID> in the request URI refers to the actual person ID or employee No.

### E.32 /ISAPI/AccessControl/CardInfo/Delete?format=json

Delete cards.

## Request URI Definition

**Table B-40 PUT /ISAPI/AccessControl/CardInfo/Delete?format=json**

<b>Method</b>	PUT
<b>Description</b>	Delete cards.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardInfoDelCond</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.33 /ISAPI/AccessControl/CardInfo/Modify?format=json

Edit card information.

## Request URI Definition

**Table B-41 PUT /ISAPI/AccessControl/CardInfo/Modify?format=json**

<b>Method</b>	PUT
<b>Description</b>	Edit card information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardInfo</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

The **employeeNo** and **cardNo** in the request message [JSON\\_CardInfo](#) cannot be edited by calling this URI. If the **cardNo** needs to be edited, you should firstly delete this card and then create a new one.

## E.34 /ISAPI/AccessControl/CardInfo/Record?format=json

Add cards and link them with a person.

**Request URI Definition****Table B-42 POST /ISAPI/AccessControl/CardInfo/Record?format=json**

<b>Method</b>	POST
<b>Description</b>	Add cards and link them with a person.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardInfo</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**E.35 /ISAPI/AccessControl/CardInfo/Search?format=json**

Search for cards.

**Request URI Definition****Table B-43 POST /ISAPI/AccessControl/CardInfo/Search?format=json**

<b>Method</b>	POST
<b>Description</b>	Search for cards.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	<u><a href="#">JSON_CardInfoSearchCond</a></u>
<b>Response</b>	<u><a href="#">JSON_CardInfoSearch</a></u>

**E.36 /ISAPI/AccessControl/CardInfo/SetUp?format=json**

Set card information.

## Request URI Definition

**Table B-44 PUT /ISAPI/AccessControl/CardInfo/SetUp?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set card information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardInfo</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

- If the device has checked that the card does not exist according to the card No., the card information will be added.
- If the device has checked that the card already exists according to the card No., the card information will be edited.
- If you want to delete a card for a person, you should set the **employeeNo** and **cardNo**, and set the **deleteCard** to "true" in the message [JSON\\_CardInfo](#). The success response message will be returned no matter whether the card exists or not. Deleting the card will only delete the card's information and will not delete the linked person information.
- If you want to delete all cards for a person, you should set the **employeeNo**, and set the **deleteCard** to "true" in the message [JSON\\_CardInfo](#). The success response message will be returned no matter whether the person exists or not or whether the person has cards or not. Deleting cards will only delete the cards' information and will not delete the linked person information.

## E.37 /ISAPI/AccessControl/CardOperations/capabilities?format=json

Get card operation capability.

## Request URI Definition

**Table B-45 GET /ISAPI/AccessControl/CardOperations/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get card operation capability.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CardOperationsCap</a></u>

	Failed: <u>JSONResponseStatus</u>
--	-----------------------------------

## E.38 /ISAPI/AccessControl/CardOperations/cardParam?format=json

Set card parameters (only available for CPU card).

### Request URI Definition

Table B-46 PUT /ISAPI/AccessControl/CardOperations/cardParam?format=json

<b>Method</b>	PUT
<b>Description</b>	Set card parameters (only available for CPU card).
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u>JSON_CardParam</u>
<b>Response</b>	<u>JSONResponseStatus</u>

## E.39 /ISAPI/AccessControl/CardOperations/clearData?format=json

Delete data from the card.

### Request URI Definition

Table B-47 PUT /ISAPI/AccessControl/CardOperations/clearData?format=json

<b>Method</b>	PUT
<b>Description</b>	Delete data from the card.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u>JSON_ClearData</u>
<b>Response</b>	Succeeded: <u>JSON_ClearDataRes</u> Failed: <u>JSONResponseStatus</u>

## E.40 /ISAPI/AccessControl/CardOperations/controlBlock?format=json

Change the control block of a specific section (only available for M1 card).

## Request URI Definition

**Table B-48 PUT /ISAPI/AccessControl/CardOperations/controlBlock?format=json**

<b>Method</b>	PUT
<b>Description</b>	Change the control block of a specific section (only available for M1 card).
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>security:</b> the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the values of the fields <b>KeyA</b> and <b>KeyB</b> should be encrypted. <b>iv:</b> the initialization vector, and it is required when <b>security</b> is 1 or 2.
<b>Request</b>	<u><a href="#">JSON_ControlBlock</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.41 /ISAPI/AccessControl/CardOperations/customData/searchTask?format=json

Search for custom card information.

## Request URI Definition

**Table B-49 POST /ISAPI/AccessControl/CardOperations/customData/searchTask?format=json**

<b>Method</b>	POST
<b>Description</b>	Search for custom card information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CustomDataSearchCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_CustomDataResult</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.42 /ISAPI/AccessControl/CardOperations/customData?format=json

Set custom card information.

### Request URI Definition

Table B-50 PUT /ISAPI/AccessControl/CardOperations/customData?format=json

<b>Method</b>	PUT
<b>Description</b>	Set custom card information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CustomData</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_CustomDataRes</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.43 /ISAPI/AccessControl/CardOperations/dataBlock/<address>?format=json

Read or write data block (only available for M1 card).

### Request URI Definition

Table B-51 GET /ISAPI/AccessControl/CardOperations/dataBlock/<address>?format=json

<b>Method</b>	GET
<b>Description</b>	Read data block (only available for M1 card).
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_DataBlock</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

Table B-52 PUT /ISAPI/AccessControl/CardOperations/dataBlock/<address>?format=json

<b>Method</b>	PUT
<b>Description</b>	Write data block (only available for M1 card).
<b>Query</b>	<b>format:</b> determine the format of request or response message.

Request	<u>JSON_DataBlock</u>
Response	<u>JSONResponseStatus</u>

**Remarks**

The <address> in the request URI refers to the block address, which is same as that in JSON\_DataBlock.

### E.44 /ISAPI/AccessControl/CardOperations/dataBlock/control?format=json

Do operations (i.e., plus, minus, copy, and paste) on the data block.

**Request URI Definition**

**Table B-53 PUT /ISAPI/AccessControl/CardOperations/dataBlock/control?format=json**

<b>Method</b>	PUT
<b>Description</b>	Do operations (i.e., plus, minus, copy, and paste) on the data block.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u>JSON_DataBlockCtrl</u>
<b>Response</b>	<u>JSONResponseStatus</u>

### E.45 /ISAPI/AccessControl/CardOperations/dataTrans?format=json

Pass through the data package (only available for CPU card).

**Request URI Definition**

**Table B-54 PUT /ISAPI/AccessControl/CardOperations/dataTrans?format=json**

<b>Method</b>	PUT
<b>Description</b>	Pass through the data package (only available for CPU card).
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u>JSON_DataTrans</u>
<b>Response</b>	<u>JSONResponseStatus</u>

## E.46 /ISAPI/AccessControl/CardOperations/encryption?format=json

Set card encryption parameters (only available for CPU card).

### Request URI Definition

**Table B-55 PUT /ISAPI/AccessControl/CardOperations/encryption?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set card encryption parameters (only available for CPU card).
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardEncryption</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u> and <b>tryTimes</b> field (card encryption attempts)

## E.47 /ISAPI/AccessControl/CardOperations/protocol?format=json

Set operation protocol type for the card (only available when applying card).

### Request URI Definition

**Table B-56 PUT /ISAPI/AccessControl/CardOperations/protocol?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set operation protocol type for the card (only available when applying card).
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardProto</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.48 /ISAPI/AccessControl/CardOperations/reset?format=json

Reset card parameters (only available for CPU card).

## Request URI Definition

**Table B-57 GET /ISAPI/AccessControl/CardOperations/reset?format=json**

<b>Method</b>	GET
<b>Description</b>	Reset card parameters (only available for CPU card).
<b>Query</b>	<b>format</b> : determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_CardResetResponse</i> Failed: <i>JSON_ResponseStatus</i>

## E.49 /ISAPI/AccessControl/CardOperations/sectionEncryption? format=json

Set the encryption parameters of a specific section (only available for M1 card).

## Request URI Definition

**Table B-58 PUT /ISAPI/AccessControl/CardOperations/sectionEncryption?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the encryption parameters of a specific section (only available for M1 card).
<b>Query</b>	<b>format</b> : determine the format of request or response message. <b>security</b> : the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the values of the fields <b>password</b> , <b>KeyA</b> , and <b>KeyB</b> should be encrypted. <b>iv</b> : the initialization vector, and it is required when <b>security</b> is 1 or 2.
<b>Request</b>	<i>JSON_SectionEncryption</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.50 /ISAPI/AccessControl/CardOperations/verification?format=json

Verify the password of the encrypted section (only available for M1 card).

### Request URI Definition

**Table B-59 PUT /ISAPI/AccessControl/CardOperations/verification?format=json**

<b>Method</b>	PUT
<b>Description</b>	Verify the password of the encrypted section (only available for M1 card).
<b>Query</b>	<p><b>format:</b> determine the format of request or response message.</p> <p><b>security:</b> the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the value of the field <b>password</b> should be encrypted.</p> <p><b>iv:</b> the initialization vector, and it is required when <b>security</b> is 1 or 2.</p>
<b>Request</b>	<u><a href="#">JSON_Verification</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.51 /ISAPI/AccessControl/CardReaderAntiSneakCfg/capabilities?format=json

Get the anti-passing back configuration capability of card readers.

### Request URI Definition

**Table B-60 GET /ISAPI/AccessControl/CardReaderAntiSneakCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the anti-passing back configuration capability of card readers.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_CardReaderAntiSneakCfg</a></u>

	Failed: <u><a href="#">JSONResponseStatus</a></u>
--	---------------------------------------------------

## E.52 /ISAPI/AccessControl/CardReaderAntiSneakCfg/<ID>?format=json

Operations about the anti-passing back configuration of a specified card reader.

### Request URI Definition

**Table B-61 GET /ISAPI/AccessControl/CardReaderAntiSneakCfg/<ID>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the anti-passing back configuration parameters of a specified card reader.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CardReaderAntiSneakCfg</a></u> Failed: <u><a href="#">JSONResponseStatus</a></u>

**Table B-62 PUT /ISAPI/AccessControl/CardReaderAntiSneakCfg/<ID>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the anti-passing back parameters of a specified card reader.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardReaderAntiSneakCfg</a></u>
<b>Response</b>	<u><a href="#">JSONResponseStatus</a></u>

### Remarks

The <ID> in the request URI refers to the card reader No.

## E.53 /ISAPI/AccessControl/CardReaderCfg/<ID>?format=json

Operations about the card reader configuration.

## Request URI Definition

**Table B-63 GET /ISAPI/AccessControl/CardReaderCfg/<ID>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the card reader configuration parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_CardReaderCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-64 PUT /ISAPI/AccessControl/CardReaderCfg/<ID>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the card reader parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_CardReaderCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## Remarks

The <ID> in the request URI refers to the card reader No. which starts from 1.

## E.54 /ISAPI/AccessControl/CardReaderCfg/capabilities?format=json

Get the configuration capability of the card reader.

## Request URI Definition

**Table B-65 GET /ISAPI/AccessControl/CardReaderCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the card reader.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_CardReaderCfg</i> Failed: <i>JSON_ResponseStatus</i>

## E.55 /ISAPI/AccessControl/CardVerificationRule/capabilities?format=json

Get the configuration capability of card No. authentication mode.

### Request URI Definition

**Table B-66 GET /ISAPI/AccessControl/CardVerificationRule/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of card No. authentication mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CardVerificationRuleCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.56 /ISAPI/AccessControl/CardVerificationRule/progress?format=json

Get the switching progress and configuration result of card No. authentication mode.

### Request URI Definition

**Table B-67 GET /ISAPI/AccessControl/CardVerificationRule/progress?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the switching progress and configuration result of card No. authentication mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CardVerificationRuleRes</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

### Remarks

- This URI is used to search for the result of switching card No. authentication (comparison) mode.
- After the card No. authentication (comparison) mode is switched, the device will check whether the card No. is duplicate.

## E.57 /ISAPI/AccessControl/CardVerificationRule?format=json

Get or set the parameters of card No. authentication mode.

### Request URI Definition

**Table B-68 GET /ISAPI/AccessControl/CardVerificationRule?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of card No. authentication mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_CardVerificationRule</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-69 PUT /ISAPI/AccessControl/CardVerificationRule?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of card No. authentication mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_CardVerificationRule</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.58 /ISAPI/AccessControl/ChannelControllerCfg

Get or set the lane controller parameters.

### Request URI Definition

**Table B-70 GET /ISAPI/AccessControl/ChannelControllerCfg**

<b>Method</b>	GET
<b>Description</b>	Get the lane controller parameters.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_ChannelControllerCfg</a></u>

	Failed: <i>XMLResponseStatus</i>
--	----------------------------------

**Table B-71 PUT /ISAPI/AccessControl/ChannelControllerCfg**

<b>Method</b>	PUT
<b>Description</b>	Set the lane controller parameters.
<b>Query</b>	None.
<b>Request</b>	<i>XML_ChannelControllerCfg</i>
<b>Response</b>	<i>XMLResponseStatus</i>

**E.59 /ISAPI/AccessControl/ChannelControllerCfg/capabilities**

Get the lane controller configuration capability.

**Request URI Definition****Table B-72 GET /ISAPI/AccessControl/ChannelControllerCfg/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the lane controller configuration capability.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>XML_Cap_ChannelControllerCfg</i> Failed: <i>XMLResponseStatus</i>

**E.60 /ISAPI/AccessControl/channelControllerTypeCfg/capabilities?format=json**

Get the configuration capability of the lane controller's device type.

**Request URI Definition****Table B-73 GET /ISAPI/AccessControl/channelControllerTypeCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the lane controller's device type.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_ChannelControllerTypeCfgCap</i> Failed: <i>JSON_ResponseStatus</i>

## E.61 /ISAPI/AccessControl/channelControllerTypeCfg?format=json

Get or set the device type parameters of the lane controller.

### Request URI Definition

Table B-74 GET /ISAPI/AccessControl/channelControllerTypeCfg?format=json

<b>Method</b>	GET
<b>Description</b>	Get the device type parameters of the lane controller.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_ChannelControllerTypeCfg</i> Failed: <i>JSON_ResponseStatus</i>

Table B-75 PUT /ISAPI/AccessControl/channelControllerTypeCfg?format=json

<b>Method</b>	PUT
<b>Description</b>	Set the device type parameters of the lane controller.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_ChannelControllerTypeCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.62 /ISAPI/AccessControl/ClearAntiSneakCfg/capabilities?format=json

Get the capability of clearing anti-passing back parameters.

## Request URI Definition

**Table B-76 GET /ISAPI/AccessControl/ClearAntiSneakCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of clearing anti-passing back parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_ClearAntiSneakCfg</i> Failed: <i>JSON_ResponseStatus</i>

## E.63 /ISAPI/AccessControl/ClearAntiSneakCfg?format=json

Clear anti-passing back parameters.

## Request URI Definition

**Table B-77 PUT /ISAPI/AccessControl/ClearAntiSneakCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Clear anti-passing back parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_ClearAntiSneakCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.64 /ISAPI/AccessControl/ClearAntiSneak?format=json

Clear anti-passing back records.

## Request URI Definition

**Table B-78 PUT /ISAPI/AccessControl/ClearAntiSneak?format=json**

<b>Method</b>	PUT
<b>Description</b>	Clear anti-passing back records.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

Request	<u>JSON_ClearAntiSneak</u>
Response	<u>JSON_ResponseStatus</u>

**E.65 /ISAPI/AccessControl/ClearAntiSneak/capabilities?format=json**

Get the capability of clearing anti-passing back records.

**Request URI Definition****Table B-79 GET /ISAPI/AccessControl/ClearAntiSneak/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of clearing anti-passing back records.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>JSON_Cap_ClearAntiSneak</u> Failed: <u>JSON_ResponseStatus</u>

**E.66 /ISAPI/AccessControl/ClearAttendancePlan?format=json**

Clear the attendance schedule.

**Request URI Definition****Table B-80 PUT /ISAPI/AccessControl/ClearAttendancePlan?format=json**

<b>Method</b>	PUT
<b>Description</b>	Clear the attendance schedule.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u>JSON_ClearAttendancePlan</u>
<b>Response</b>	<u>JSON_ResponseStatus</u>

**E.67 /ISAPI/AccessControl/ClearCardRecord**

Clear card swiping records in the cross-controller anti-passing back server.

## Request URI Definition

**Table B-81 PUT /ISAPI/AccessControl/ClearCardRecord**

<b>Method</b>	PUT
<b>Description</b>	Clear card swiping records in the cross-controller anti-passing back server.
<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML_ClearCardRecord</a></u>
<b>Response</b>	<u><a href="#">XML_ResponseStatus</a></u>

## Remarks

This request URI can only be used by the cross-controller anti-passing back server, and it is not supported by the cross-controller anti-passing back devices based on card mode.

## E.68 /ISAPI/AccessControl/ClearCardRecord/capabilities

Get the capability of clearing card swiping records in the cross-controller anti-passing back server.

## Request URI Definition

**Table B-82 GET /ISAPI/AccessControl/ClearCardRecord/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the capability of clearing card swiping records in the cross-controller anti-passing back server.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_ClearCardRecord</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

## E.69 /ISAPI/AccessControl/ClearPictureCfg/capabilities?format=json

Get the capability of clearing all pictures in the device.

## Request URI Definition

**Table B-83 GET /ISAPI/AccessControl/ClearPictureCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of clearing all pictures in the device.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_ClearPictureCfgCap</i> Failed: <i>JSON_ResponseStatus</i>

## E.70 /ISAPI/AccessControl/ClearPictureCfg?format=json

Clear all pictures in the device.

## Request URI Definition

**Table B-84 PUT /ISAPI/AccessControl/ClearPictureCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Clear all pictures in the device.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_ClearPictureCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.71 /ISAPI/AccessControl/ClearPlansCfg/capabilities?format=json

Get the capability of clearing access control schedule configuration.

## Request URI Definition

**Table B-85 GET /ISAPI/AccessControl/ClearPlansCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of clearing access control schedule configuration.

<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_Cap_ClearPlansCfg</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

**Remarks**

This URI is not supported by integration of information release system.

**E.72 /ISAPI/AccessControl/ClearPlansCfg?format=json**

Clear the access control schedule configuration.

**Request URI Definition**

**Table B-86 PUT /ISAPI/AccessControl/ClearPlansCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Clear the access control schedule configuration parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<a href="#"><u>JSON_ClearPlansCfg</u></a>
<b>Response</b>	<a href="#"><u>JSON_ResponseStatus</u></a>

**E.73 /ISAPI/AccessControl/ClearSubmarineBack**

Clear cross-controller anti-passing back parameters.

**Request URI Definition****Table B-87 PUT /ISAPI/AccessControl/ClearSubmarineBack**

<b>Method</b>	PUT
<b>Description</b>	Clear cross-controller anti-passing back parameters.
<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML_ClearSubmarineBack</a></u>
<b>Response</b>	<u><a href="#">XML_ResponseStatus</a></u>

**E.74 /ISAPI/AccessControl/ClearSubmarineBack/capabilities**

Get the capability of clearing cross-controller anti-passing back parameters.

**Request URI Definition****Table B-88 GET /ISAPI/AccessControl/ClearSubmarineBack/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the capability of clearing cross-controller anti-passing back parameters.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_ClearSubmarineBack</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

**E.75 /ISAPI/AccessControl/Configuration/attendanceMode/capabilities?format=json**

Get the configuration capability of the attendance mode.

**Request URI Definition****Table B-89 GET /ISAPI/AccessControl/Configuration/attendanceMode/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the attendance mode.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_AttendanceMode</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.76 /ISAPI/AccessControl/Configuration/attendanceMode?format=json

Get or set the attendance mode parameters.

### Request URI Definition

Table B-90 GET /ISAPI/AccessControl/Configuration/attendanceMode?format=json

<b>Method</b>	GET
<b>Description</b>	Get the attendance mode parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_AttendanceMode</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

Table B-91 PUT /ISAPI/AccessControl/Configuration/attendanceMode?format=json

<b>Method</b>	PUT
<b>Description</b>	Set the attendance mode parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_AttendanceMode</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.77 /ISAPI/AccessControl/Configuration/IRCfg/capabilities?format=json

Get active infrared intrusion capability.

## Request URI Definition

**Table B-92 GET /ISAPI/AccessControl/Configuration/IRCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get active infrared intrusion capability.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_IRCfgCap</i> Failed: <i>JSON_ResponseStatus</i>

## E.78 /ISAPI/AccessControl/Configuration/IRCfg?format=json

Get or set active infrared intrusion parameters.

## Request URI Definition

**Table B-93 GET /ISAPI/AccessControl/Configuration/IRCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get active infrared intrusion parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_IRCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-94 PUT /ISAPI/AccessControl/Configuration/IRCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set active infrared intrusion parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_IRCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.79 /ISAPI/AccessControl/Configuration/NFCCfg/capabilities?format=json

Get the configuration capability of enabling NFC (Near-Field Communication) function.

### Request URI Definition

**Table B-95 GET /ISAPI/AccessControl/Configuration/NFCCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of enabling NFC (Near-Field Communication) function.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_NFCCfgCap</i> Failed: <i>JSON_ResponseStatus</i>

## E.80 /ISAPI/AccessControl/Configuration/NFCCfg?format=json

Operations about the configuration of enabling NFC (Near-Field Communication) function.

### Request URI Definition

**Table B-96 GET /ISAPI/AccessControl/Configuration/NFCCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of enabling NFC (Near-Field Communication) function.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_NFCCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-97 PUT /ISAPI/AccessControl/Configuration/NFCCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of enabling NFC (Near-Field Communication) function.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_NFCCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.81 /ISAPI/AccessControl/Configuration/RFCardCfg/capabilities?format=json

Get the configuration capability of enabling RF (Radio Frequency) card recognition.

### Request URI Definition

**Table B-98 GET /ISAPI/AccessControl/Configuration/RFCardCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of enabling RF (Radio Frequency) card recognition.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_RFCardCfgCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.82 /ISAPI/AccessControl/Configuration/RFCardCfg?format=json

Operations about the configuration of enabling RF (Radio Frequency) card recognition.

## Request URI Definition

**Table B-99 GET /ISAPI/AccessControl/Configuration/RFCardCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of enabling RF (Radio Frequency) card recognition.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_RFCardCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-100 PUT /ISAPI/AccessControl/Configuration/RFCardCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of enabling RF (Radio Frequency) card recognition.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_RFCardCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.83 /ISAPI/AccessControl/Configuration/safetyHelmetDetection/capabilities?format=json

Get the configuration capability of hard hat detection.

## Request URI Definition

**Table B-101 GET /ISAPI/AccessControl/Configuration/safetyHelmetDetection/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of hard hat detection.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_SafetyHelmetDetectionCap</a></u>

	Failed: <i>JSONResponseStatus</i>
--	-----------------------------------

## E.84 /ISAPI/AccessControl/Configuration/safetyHelmetDetection?format=json

Get or set the parameters of hard hat detection.

### Request URI Definition

**Table B-102 GET /ISAPI/AccessControl/Configuration/safetyHelmetDetection?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of hard hat detection.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_SafetyHelmetDetection</i> Failed: <i>JSONResponseStatus</i>

**Table B-103 PUT /ISAPI/AccessControl/Configuration/safetyHelmetDetection?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of hard hat detection.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_SafetyHelmetDetection</i>
<b>Response</b>	<i>JSONResponseStatus</i>

## E.85 /ISAPI/AccessControl/DeployInfo

Get the arming information (e.g., arming types).

### Request URI Definition

**Table B-104 GET /ISAPI/AccessControl/DeployInfo**

<b>Method</b>	GET
<b>Description</b>	Get the arming information (e.g., arming types).

<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>XML_DeployInfo</i> Failed: <i>XML_ResponseStatus</i>

**Remarks**

The client arming supports arming of only one channel and can upload offline events. The real-time arming is used for other devices to arm the access control devices, which supports arming of up to four channels and cannot upload offline events.

**E.86 /ISAPI/AccessControl/DeployInfo/capabilities**

Get the capability of getting arming information.

**Request URI Definition****Table B-105 GET /ISAPI/AccessControl/DeployInfo/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the capability of getting arming information.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>XML_Cap_DeployInfo</i> Failed: <i>XML_ResponseStatus</i>

**E.87 /ISAPI/AccessControl/EventOptimizationCfg/capabilities?format=json**

Get the configuration capability of event optimization.

**Request URI Definition****Table B-106 GET /ISAPI/AccessControl/EventOptimizationCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of event optimization.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_EventOptimizationCfg</i> Failed: <i>JSON_ResponseStatus</i>

## E.88 /ISAPI/AccessControl/EventOptimizationCfg?format=json

Operations about the event optimization configuration.

### Request URI Definition

Table B-107 GET /ISAPI/AccessControl/EventOptimizationCfg?format=json

<b>Method</b>	GET
<b>Description</b>	Get the event optimization configuration parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_EventOptimizationCfg</i> Failed: <i>JSON_ResponseStatus</i>

Table B-108 PUT /ISAPI/AccessControl/EventOptimizationCfg?format=json

<b>Method</b>	PUT
<b>Description</b>	Set the event optimization parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_EventOptimizationCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.89 /ISAPI/AccessControl/FaceCompareCond

Get or set the condition parameters of face picture comparison.

## Request URI Definition

**Table B-109 GET /ISAPI/AccessControl/FaceCompareCond**

<b>Method</b>	GET
<b>Description</b>	Get the condition parameters of face picture comparison.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>XML_FaceCompareCond</i> Failed: <i>XMLResponseStatus</i>

**Table B-110 PUT /ISAPI/AccessControl/FaceCompareCond**

<b>Method</b>	PUT
<b>Description</b>	Set the condition parameters of face picture comparison.
<b>Query</b>	None.
<b>Request</b>	<i>XML_FaceCompareCond</i>
<b>Response</b>	<i>XMLResponseStatus</i>

## E.90 /ISAPI/AccessControl/FaceCompareCond/capabilities

Get condition configuration capability of face picture comparison.

## Request URI Definition

**Table B-111 GET /ISAPI/AccessControl/FaceCompareCond/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get condition configuration capability of face picture comparison.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>XML_Cap_FaceCompareCond</i> Failed: <i>XMLResponseStatus</i>

## E.91 /ISAPI/AccessControl/FaceRecognizeMode/capabilities?format=json

Get the configuration capability of the facial recognition mode.

### Request URI Definition

**Table B-112 GET /ISAPI/AccessControl/FaceRecognizeMode/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the facial recognition mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_Cap_FaceRecognizeMode</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

### Remarks

Switching facial recognition mode will clear face permission information in the device.

## E.92 /ISAPI/AccessControl/FaceRecognizeMode?format=json

Operations about the configuration of the facial recognition mode.

### Request URI Definition

**Table B-113 GET /ISAPI/AccessControl/FaceRecognizeMode?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of the facial recognition mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_FaceRecognizeMode</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

**Table B-114 PUT /ISAPI/AccessControl/FaceRecognizeMode?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the facial recognition mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_FaceRecognizeMode</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**Remarks**

Switching facial recognition mode will clear face permission information in the device.

### **E.93 /ISAPI/AccessControl/FaceTemperatureEvent/capabilities?format=json**

Get the capability of actively getting face temperature screening events.

**Request URI Definition****Table B-115 GET /ISAPI/AccessControl/FaceTemperatureEvent/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of actively getting face temperature screening events.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_FaceTemperatureEventCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

### **E.94 /ISAPI/AccessControl/FaceTemperatureEvent?format=json**

Get face temperature screening events actively.

## Request URI Definition

**Table B-116 POST /ISAPI/AccessControl/FaceTemperatureEvent?format=json**

<b>Method</b>	POST
<b>Description</b>	Get face temperature screening events actively.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_FaceTemperatureEventCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_FaceTemperatureEvent</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.95 /ISAPI/AccessControl/FingerPrint/DeleteProcess?format=json

Get the progress of deleting fingerprint data.

## Request URI Definition

**Table B-117 GET /ISAPI/AccessControl/FingerPrint/DeleteProcess?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the progress of deleting fingerprint data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_FingerPrintDeleteProcess</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

When starting deleting fingerprint data, this URI will be repeatedly called to get the deleting progress until "success" or "failed" is returned by the parameter **status** in the message [JSON\\_FingerPrintDeleteProcess](#).

## E.96 /ISAPI/AccessControl/FingerPrint/Delete/capabilities?format=json

Get the capability of deleting fingerprint data.

## Request URI Definition

**Table B-118 GET /ISAPI/AccessControl/FingerPrint/Delete/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of deleting fingerprint data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_FingerPrintDelete</i> Failed: <i>JSON_ResponseStatus</i>

## E.97 /ISAPI/AccessControl/FingerPrint/Delete?format=json

Start deleting the fingerprint data.

## Request URI Definition

**Table B-119 PUT /ISAPI/AccessControl/FingerPrint/Delete?format=json**

<b>Method</b>	PUT
<b>Description</b>	Start deleting the fingerprint data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_FingerPrintDelete</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## Remarks

This URI is only used to start deleting. To judge whether the deleting is completed, you should call the request URI */ISAPI/AccessControl/FingerPrint/DeleteProcess?format=json* by GET method to get the deleting status.

## E.98 /ISAPI/AccessControl/FingerPrint/SetUp?format=json

Set the fingerprint parameters.

## Request URI Definition

**Table B-120 POST /ISAPI/AccessControl/FingerPrint/SetUp?format=json**

<b>Method</b>	POST
<b>Description</b>	Set the fingerprint parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_FingerPrintCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_FingerPrintStatus</a></u>

## Remarks

- If the **fingerData** is not applied, it indicates editing fingerprint parameters instead of applying fingerprint data to the fingerprint module.
- If the **fingerData** is applied, the fingerprint data will be added if it does not exist in the fingerprint module, or the original fingerprint data will be overwritten if it already exists in the fingerprint module.
- There are four different methods for deleting one or more fingerprints:
  - To delete a specific fingerprint in a specific fingerprint module linked with a specific employee No., the **employeeNo**, **enableCardReader**, **fingerPrintID**, and **deleteFingerPrint** in the message [JSON\\_FingerPrintCfg](#) should be configured, and the success response message will be returned no matter whether the fingerprint exists or not.
  - To delete a specific fingerprint in all fingerprint modules linked with a specific employee No., the **employeeNo**, **fingerPrintID**, and **deleteFingerPrint** in the message [JSON\\_FingerPrintCfg](#) should be configured, and the success response message will be returned no matter whether the fingerprints exist or not.
  - To delete all fingerprints in a specific fingerprint module linked with a specific employee No., the **employeeNo**, **enableCardReader**, and **deleteFingerPrint** in the message [JSON\\_FingerPrintCfg](#) should be configured, and the success response message will be returned no matter whether the fingerprints exist or not.
  - To delete all fingerprints in all fingerprint modules linked with a specific employee No., the **employeeNo** and **deleteFingerPrint** in the message [JSON\\_FingerPrintCfg](#) should be configured, and the success response message will be returned no matter whether the fingerprints exist or not.

## E.99 /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json

Get the configuration capability of fingerprint parameters.

## Request URI Definition

**Table B-121 GET /ISAPI/AccessControl/FingerPrintCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of fingerprint parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_FingerPrintCfg</i> Failed: <i>JSON_ResponseStatus</i>

## E.100 /ISAPI/AccessControl/FingerPrintDownload?format=json

Set fingerprint parameters to link with a person, and apply the collected fingerprint data.

## Request URI Definition

**Table B-122 POST /ISAPI/AccessControl/FingerPrintDownload?format=json**

<b>Method</b>	POST
<b>Description</b>	Set fingerprint parameters to link with a person, and apply the collected fingerprint data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_FingerPrintCfg</i> +fingerprint data (by boundary method)
<b>Response</b>	<i>JSON_ResponseStatus</i>

## Remarks

This URI is only used to start applying the fingerprint data. To check whether the applying is completed, you should call the request URI **/ISAPI/AccessControl/FingerPrintProgress?format=json** by GET method to get the applying status.

## E.101 /ISAPI/AccessControl/FingerPrintModify?format=json

Edit fingerprint parameters.

## Request URI Definition

**Table B-123 POST /ISAPI/AccessControl/FingerPrintModify?format=json**

<b>Method</b>	POST
<b>Description</b>	Edit fingerprint parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_FingerPrintModify</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

### Remarks

Only the fingerprint parameters can be edited. The collected fingerprint data will not be edited and applied.

## E.102 /ISAPI/AccessControl/FingerPrintProgress?format=json

Get the progress of applying fingerprint data.

## Request URI Definition

**Table B-124 GET /ISAPI/AccessControl/FingerPrintProgress?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the progress of applying fingerprint data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_FingerPrintStatus</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

### Remarks

When starting applying fingerprint data, this URI will be repeatedly called to get the applying progress until "1" is returned by the parameter **totalStatus** in the message

[JSON\\_FingerPrintStatus](#).

## E.103 /ISAPI/AccessControl/FingerPrintUpload?format=json

Get the fingerprint information, including fingerprint parameters and data.

## Request URI Definition

**Table B-125 POST /ISAPI/AccessControl/FingerPrintUpload?format=json**

<b>Method</b>	POST
<b>Description</b>	Get the fingerprint information, including fingerprint parameters and data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<b><u>JSON_FingerPrintCond</u></b>
<b>Response</b>	<b><u>JSON_FingerPrintInfo</u></b> +fingerprint data (by boundary method)

## Remarks

- To get the information of a specific fingerprint, the **searchID**, **employeeNo**, **cardReaderNo**, and **fingerPrintID** in the message **JSON\_FingerPrintCond** should be configured. If the fingerprint matching the search conditions exists, the **status** will be set to "OK" and the corresponding fingerprint information will be returned by **FingerPrintList** in the message **JSON\_FingerPrintInfo**; otherwise, the **status** will be set to "NoFP" and the **FingerPrintList** will be set to NULL in the message **JSON\_FingerPrintInfo**.
- To get all fingerprints linked with a specific employee No. (person ID), the **searchID** and **employeeNo** in the message **JSON\_FingerPrintCond** should be configured. If the fingerprints matching the search conditions exist, the **status** will be set to "OK" and the corresponding fingerprint information will be returned by **FingerPrintList** in the message **JSON\_FingerPrintInfo**. The request URI **/ISAPI/AccessControl/FingerPrintUpload?format=json** will be repeatedly called by POST method to get the information of multiple fingerprints matching the search conditions until "NoFP" is returned by **status** in the message **JSON\_FingerPrintInfo** (it indicates that information of all fingerprints matching the search conditions are obtained). If there is no fingerprint matching the search conditions, the **status** will be set to "NoFP" and the **FingerPrintList** will be set to NULL in the message **JSON\_FingerPrintInfo**.

## E.104 /ISAPI/AccessControl/GetAcsEvent/capabilities

Get capability of getting access control event.

## Request URI Definition

**Table B-126 GET /ISAPI/AccessControl/GetAcsEvent/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get capability of getting access control event.

<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	<u><a href="#">XML_Cap_GetAcsEvent</a></u>

## E.105 /ISAPI/AccessControl/healthCodeCfg/capabilities?format=json

Get the configuration capability of the health code.

### Request URI Definition

**Table B-127 GET /ISAPI/AccessControl/healthCodeCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the health code.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_HealthCodeCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

### Remarks

The device needs to connect to the health code server for uploading the information of the person to be authenticated to the server. The health code server is provided by the third party.

## E.106 /ISAPI/AccessControl/healthCodeCfg?format=json

Get or set the health code parameters.

### Request URI Definition

**Table B-128 GET /ISAPI/AccessControl/healthCodeCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the health code parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_HealthCodeCfg</a></u>

	Failed: <i>JSONResponseStatus</i>
--	-----------------------------------

**Table B-129 PUT /ISAPI/AccessControl/healthCodeCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the health code parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_HealthCodeCfg</i>
<b>Response</b>	<i>JSONResponseStatus</i>

**E.107 /ISAPI/AccessControl/IDCardInfoEvent/capabilities?format=json**

Get the capability of getting the ID card swiping events actively.

**Request URI Definition****Table B-130 GET /ISAPI/AccessControl/IDCardInfoEvent/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of getting the ID card swiping events actively.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_IDCardInfoEventCap</i> Failed: <i>JSONResponseStatus</i>

**E.108 /ISAPI/AccessControl/IDCardInfoEvent?format=json**

Get the ID card swiping events actively.

**Request URI Definition****Table B-131 POST /ISAPI/AccessControl/IDCardInfoEvent?format=json**

<b>Method</b>	POST
<b>Description</b>	Get the ID card swiping events actively.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

Request	<u>JSON_IDCardInfoEventCond</u>
Response	<u>JSON_IDCardInfoEvent</u>

## E.109 /ISAPI/AccessControl/IdentityTerminal

Operations about configuration of intelligent identity recognition terminal.

### Request URI Definition

**Table B-132 GET /ISAPI/AccessControl/IdentityTerminal**

<b>Method</b>	GET
<b>Description</b>	Get the configuration parameters of intelligent identity recognition terminal.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>XML_IdentityTerminal</u> Failed: <u>XML_ResponseStatus</u>

**Table B-133 PUT /ISAPI/AccessControl/IdentityTerminal**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of intelligent identity recognition terminal.
<b>Query</b>	None.
<b>Request</b>	<u>XML_IdentityTerminal</u>
<b>Response</b>	<u>XML_ResponseStatus</u>

## E.110 /ISAPI/AccessControl/IdentityTerminal/capabilities

Get configuration capability of intelligent identity recognition terminal.

## Request URI Definition

**Table B-134 GET /ISAPI/AccessControl/IdentityTerminal/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get configuration capability of intelligent identity recognition terminal.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_IdentityTerminal</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

## E.111 /ISAPI/AccessControl/keyCfg/<ID>/attendance?format=json

Get or set the parameters of attendance check by pressing the key.

## Request URI Definition

**Table B-135 GET /ISAPI/AccessControl/keyCfg/<ID>/attendance?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of attendance check by pressing the key.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Attendance</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-136 PUT /ISAPI/AccessControl/keyCfg/<ID>/attendance?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of attendance check by pressing the key.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_Attendance</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.112 /ISAPI/AccessControl/keyCfg/attendance/capabilities? format=json

Get the configuration capability of attendance check by pressing the key.

### Request URI Definition

**Table B-137 GET /ISAPI/AccessControl/keyCfg/attendance/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of attendance check by pressing the key.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_AttendanceCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.113 /ISAPI/AccessControl/keyCfg/attendance?format=json

Get the attendance parameter list.

### Request URI Definition

**Table B-138 GET /ISAPI/AccessControl/keyCfg/attendance?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the attendance parameter list.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_AttendanceList</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.114 /ISAPI/AccessControl/LogModeCfg/capabilities?format=json

Get the configuration capability of the log mode.

## Request URI Definition

**Table B-139 GET /ISAPI/AccessControl/LogModeCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the log mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_LogModeCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.115 /ISAPI/AccessControl/LogModeCfg?format=json

Operations about the log mode configuration.

## Request URI Definition

**Table B-140 GET /ISAPI/AccessControl/LogModeCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the log mode configuration parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_LogModeCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-141 PUT /ISAPI/AccessControl/LogModeCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the log mode parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_LogModeCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.116 /ISAPI/AccessControl/maskDetection/capabilities?format=json

Get the configuration capability of mask detection.

### Request URI Definition

**Table B-142 GET /ISAPI/AccessControl/maskDetection/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of mask detection.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_MaskDetectionCap</i> Failed: <i>JSON_ResponseStatus</i>

## E.117 /ISAPI/AccessControl/maskDetection?format=json

Get or set the mask detection parameters.

### Request URI Definition

**Table B-143 GET /ISAPI/AccessControl/maskDetection?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the mask detection parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_MaskDetection</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-144 PUT /ISAPI/AccessControl/maskDetection?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the mask detection parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

Request	<u>JSON_MaskDetection</u>
Response	<u>JSON_ResponseStatus</u>

**E.118 /ISAPI/AccessControl/OfflineCapture/capabilities?format=json**

Get the offline collection capability.

**Request URI Definition****Table B-145 GET /ISAPI/AccessControl/OfflineCapture/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the offline collection capability.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>JSON_OfflineCaptureCap</u> Failed: <u>JSON_ResponseStatus</u>

**E.119 /ISAPI/AccessControl/OfflineCapture/DataCollections/<captureNo>?format=json**

Deleted a specific piece of offline collected data.

**Request URI Definition****Table B-146 DELETE /ISAPI/AccessControl/OfflineCapture/DataCollections/<captureNo>?format=json**

<b>Method</b>	DELETE
<b>Description</b>	Deleted a specific piece of offline collected data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	<u>JSON_ResponseStatus</u>

**Remarks**

The <captureNo> in the request URI refers to the collection No.

## E.120 /ISAPI/AccessControl/OfflineCapture/DataCollections/searchTask?format=json

Search for the collected data.

### Request URI Definition

Table B-147 POST /ISAPI/AccessControl/OfflineCapture/DataCollections/searchTask?format=json

<b>Method</b>	POST
<b>Description</b>	Search for the collected data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_SearchTaskCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_SearchTaskResponse</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.121 /ISAPI/AccessControl/OfflineCapture/DataCollections?format=json

Delete all offline collected data.

### Request URI Definition

Table B-148 DELETE /ISAPI/AccessControl/OfflineCapture/DataCollections?format=json

<b>Method</b>	DELETE
<b>Description</b>	Delete all offline collected data.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.122 /ISAPI/AccessControl/OfflineCapture/dataOutput/progress?format=json

Get the progress of exporting the offline collected data.

## Request URI Definition

**Table B-149 GET /ISAPI/AccessControl/OfflineCapture/dataOutput/progress?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the progress of exporting the offline collected data.
<b>Query</b>	<b>format</b> : determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_DataOutputProgress</i> Failed: <i>JSON_ResponseStatus</i>

## E.123 /ISAPI/AccessControl/OfflineCapture/dataOutput?format=json

Export the offline collected data.

## Request URI Definition

**Table B-150 PUT /ISAPI/AccessControl/OfflineCapture/dataOutput?format=json**

<b>Method</b>	PUT
<b>Description</b>	Export the offline collected data.
<b>Query</b>	<b>format</b> : determine the format of request or response message. <b>security</b> : the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. In the message of this request URI, the value of the field <b>password</b> should be encrypted. <b>iv</b> : the initialization vector, and it is required when <b>security</b> is 1 or 2.
<b>Request</b>	<i>JSON_DataOutputCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.124 /ISAPI/AccessControl/OfflineCapture/progress?format=json

Get the offline collection progress.

## Request URI Definition

**Table B-151 GET /ISAPI/AccessControl/OfflineCapture/progress?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the offline collection progress.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_CaptureProgress</i> Failed: <i>JSON_ResponseStatus</i>

## E.125 /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json

Get or set the parameters of offline collection rules.

## Request URI Definition

**Table B-152 GET /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of offline collection rules.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_RuleInfo</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-153 PUT /ISAPI/AccessControl/OfflineCapture/ruleInfo?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of offline collection rules.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_RuleInfo</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.126 /ISAPI/AccessControl/OfflineCapture/uploadFailedDetails? format=json

Get the details of failing to upload the user list of offline collection.

### Request URI Definition

**Table B-154 GET /ISAPI/AccessControl/OfflineCapture/uploadFailedDetails?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the details of failing to upload the user list of offline collection.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON UploadFailedDetails</a></u> Failed: <u><a href="#">JSON ResponseStatus</a></u>

## E.127 /ISAPI/AccessControl/OSDPMModify/<ID>?format=json

Set the OSDP (Open Supervised Device Protocol) card reader ID.

### Request URI Definition

**Table B-155 PUT /ISAPI/AccessControl/OSDPMModify/<ID>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the OSDP (Open Supervised Device Protocol) card reader ID.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON OSDPModify</a></u>
<b>Response</b>	<u><a href="#">JSON ResponseStatus</a></u>

### Remarks

The <ID> in the request URI refers to the original OSDP card reader ID which is between 0 and 126, and 127 refers to broadcast.

## E.128 /ISAPI/AccessControl/OSDPMModify/capabilities?format=json

Get the capability of editing the OSDP (Open Supervised Device Protocol) card reader ID.

## Request URI Definition

**Table B-156 GET /ISAPI/AccessControl/OSDPMModify/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of editing the OSDP (Open Supervised Device Protocol) card reader ID.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap OSDPModify</a></u> Failed: <u><a href="#">JSONResponseStatus</a></u>

## E.129 /ISAPI/AccessControl/OSDPStatus/<ID>?format=json

Get the OSDP (Open Supervised Device Protocol) card reader status.

## Request URI Definition

**Table B-157 GET /ISAPI/AccessControl/OSDPStatus/<ID>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the OSDP (Open Supervised Device Protocol) card reader status.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_OSDPStatus</a></u> Failed: <u><a href="#">JSONResponseStatus</a></u>

## Remarks

The <ID> in the request URI refers to the OSDP card reader ID which is between 0 and 126, and 127 refers to broadcast. Limited by the device, the OSDP card reader status can only be obtained one by one.

## E.130 /ISAPI/AccessControl/OSDPStatus/capabilities?format=json

Get the capability of getting the OSDP (Open Supervised Device Protocol) card reader status.

## Request URI Definition

**Table B-158 GET /ISAPI/AccessControl/OSDPStatus/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of getting the OSDP (Open Supervised Device Protocol) card reader status.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_OSDPStatus</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.131 /ISAPI/AccessControl/personInfoExtendName/capabilities?format=json

Get the configuration capability of the name of the additional person information.

## Request URI Definition

**Table B-159 GET /ISAPI/AccessControl/personInfoExtendName/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the name of the additional person information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_PersonInfoExtendNameCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.132 /ISAPI/AccessControl/personInfoExtendName?format=json

Get or set the parameters of the name of the additional person information.

## Request URI Definition

**Table B-160 GET /ISAPI/AccessControl/personInfoExtendName?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of the name of the additional person information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_PersonInfoExtendName</a></u> Failed: <u><a href="#">JSONResponseStatus</a></u>

**Table B-161 PUT /ISAPI/AccessControl/personInfoExtendName?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the name of the additional person information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_PersonInfoExtendName</a></u>
<b>Response</b>	<u><a href="#">JSONResponseStatus</a></u>

## E.133 /ISAPI/AccessControl/QRCodeEvent/capabilities?format=json

Get the capability of actively getting QR code scanning events.

## Request URI Definition

**Table B-162 GET /ISAPI/AccessControl/QRCodeEvent/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of actively getting QR code scanning events.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_QRCodeEventCap</a></u> Failed: <u><a href="#">JSONResponseStatus</a></u>

**E.134 /ISAPI/AccessControl/QRCodeEvent?format=json**

Get QR code scanning events actively.

**Request URI Definition****Table B-163 POST /ISAPI/AccessControl/QRCodeEvent?format=json**

<b>Method</b>	POST
<b>Description</b>	Get QR code scanning events actively.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_QRCodeEventCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_QRCodeEvent</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**E.135 /ISAPI/AccessControl/ReaderAcrossHost**

Operations about the cross-controller anti-passing back configuration of card readers.

**Request URI Definition****Table B-164 GET /ISAPI/AccessControl/ReaderAcrossHost**

<b>Method</b>	GET
<b>Description</b>	Get the cross-controller anti-passing back parameters of card readers.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_ReaderAcrossHost</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

**Table B-165 PUT /ISAPI/AccessControl/ReaderAcrossHost**

<b>Method</b>	PUT
<b>Description</b>	Set the cross-controller anti-passing back parameters of card readers.
<b>Query</b>	None.

Request	<u>XML_ReaderAcrossHost</u>
Response	<u>XML_ResponseStatus</u>

### E.136 /ISAPI/AccessControl/ReaderAcrossHost/capabilities

Get the configuration capability of cross-controller anti-passing back status of card readers.

#### Request URI Definition

Table B-166 GET /ISAPI/AccessControl/ReaderAcrossHost/capabilities

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of cross-controller anti-passing back status of card readers.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>XML_Cap_ReaderAcrossHost</u> Failed: <u>XML_ResponseStatus</u>

### E.137 /ISAPI/AccessControl/remoteCheck/capabilities?format=json

Get the capability of verifying the access control event remotely.

#### Request URI Definition

Table B-167 GET /ISAPI/AccessControl/remoteCheck/capabilities?format=json

<b>Method</b>	GET
<b>Description</b>	Get the capability of verifying the access control event remotely.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>JSON_Cap_RemoteCheck</u> Failed: <u>JSON_ResponseStatus</u>

**E.138 /ISAPI/AccessControl/remoteCheck?format=json**

Verify the access control event remotely.

**Request URI Definition****Table B-168 PUT /ISAPI/AccessControl/remoteCheck?format=json**

<b>Method</b>	PUT
<b>Description</b>	Verify the access control event remotely to control opening or closing the door.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_RemoteCheck</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**E.139 /ISAPI/AccessControl/RemoteControl/buzzer/<ID>?format=json**

Remotely control the buzzer of the card reader.

**Request URI Definition****Table B-169 PUT /ISAPI/AccessControl/RemoteControl/buzzer/<ID>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Remotely control the buzzer of the card reader.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_RemoteControlBuzzer</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**Remarks**

The <ID> in the request URI refers to the buzzer No., which is also the No. of the card reader. If the <ID> is 65535, it refers to all buzzers (card readers).

**E.140 /ISAPI/AccessControl/RemoteControl/buzzer/capabilities?format=json**

Get the capability of remotely controlling the buzzer of the card reader.

## Request URI Definition

**Table B-170 GET /ISAPI/AccessControl/RemoteControl/buzzer/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the capability of remotely controlling the buzzer of the card reader.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_RemoteControlBuzzer</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.141 /ISAPI/AccessControl/remoteControllerModeCfg/capabilities?format=json

Get the configuration capability of the keyfob control mode.

## Request URI Definition

**Table B-171 GET /ISAPI/AccessControl/remoteControllerModeCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the keyfob control mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_RemoteControllerModeCfgCap</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.142 /ISAPI/AccessControl/remoteControllerModeCfg?format=json

Get or set the parameters of the keyfob control mode.

## Request URI Definition

**Table B-172 GET /ISAPI/AccessControl/remoteControllerModeCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of the keyfob control mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_RemoteControllerModeCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-173 PUT /ISAPI/AccessControl/remoteControllerModeCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the keyfob control mode.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_RemoteControllerModeCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.143 /ISAPI/AccessControl/ServerDevice

Operation about the configuration of cross-controller anti-passing back server information.

## Request URI Definition

**Table B-174 GET /ISAPI/AccessControl/ServerDevice**

<b>Method</b>	GET
<b>Description</b>	Get the information (i.e., IP address and port No.) of the cross-controller anti-passing back server.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>XML_ServerDevice</i> Failed: <i>XML_ResponseStatus</i>

**Table B-175 PUT /ISAPI/AccessControl/ServerDevice**

<b>Method</b>	PUT
<b>Description</b>	Set the information (i.e., IP address and port No.) of the cross-controller anti-passing back server.
<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML_ServerDevice</a></u>
<b>Response</b>	<u><a href="#">XML_ResponseStatus</a></u>

**E.144 /ISAPI/AccessControl/ServerDevice/capabilities**

Get the configuration capability of cross-controller anti-passing back server information.

**Request URI Definition****Table B-176 GET /ISAPI/AccessControl/ServerDevice/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of cross-controller anti-passing back server information.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_ServerDevice</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

**E.145 /ISAPI/AccessControl/showHealthCodeCfg/capabilities?format=json**

Get the configuration capability of health code display parameters.

**Request URI Definition****Table B-177 GET /ISAPI/AccessControl/showHealthCodeCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of health code display parameters.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_HealthCodeDisplayCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Remarks**

When the device does not display the health code for privacy reasons after getting it, whether to open the door for the person is determined by the health code type.

**E.146 /ISAPI/AccessControl/showHealthCodeCfg?format=json**

Get or set the health code display parameters.

**Request URI Definition****Table B-178 GET /ISAPI/AccessControl/showHealthCodeCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the health code display parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_HealthCodeDisplayCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-179 PUT /ISAPI/AccessControl/showHealthCodeCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the health code display parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_HealthCodeDisplayCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

**E.147 /ISAPI/AccessControl/SubmarineBackMode**

Operations about the configuration of cross-controller anti-passing back mode and rule.

## Request URI Definition

**Table B-180 GET /ISAPI/AccessControl/SubmarineBackMode**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of cross-controller anti-passing back mode and rule.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>XML_SubmarineBackMode</u> Failed: <u>XML_ResponseStatus</u>

**Table B-181 PUT /ISAPI/AccessControl/SubmarineBackMode**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of cross-controller anti-passing back mode and rule.
<b>Query</b>	None.
<b>Request</b>	<u>XML_SubmarineBackMode</u>
<b>Response</b>	<u>XML_ResponseStatus</u>

## E.148 /ISAPI/AccessControl/SubmarineBackMode/capabilities

Get the configuration capability of cross-controller anti-passing back mode and rule.

## Request URI Definition

**Table B-182 GET /ISAPI/AccessControl/SubmarineBackMode/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of cross-controller anti-passing back mode and rule.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>XML_Cap_SubmarineBackMode</u> Failed: <u>XML_ResponseStatus</u>

**E.149 /ISAPI/AccessControl/SubmarineBackReader/capabilities**

Get the configuration capability of card readers for cross-controller anti-passing back.

**Request URI Definition****Table B-183 GET /ISAPI/AccessControl/SubmarineBackReader/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of card readers for cross-controller anti-passing back.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>XML_Cap_SubmarineBackReader</u></a> Failed: <a href="#"><u>XML_ResponseStatus</u></a>

**E.150 /ISAPI/AccessControl/SubmarineBackReader/ConfigureNo/<ID>**

Operations about the configuration of card readers for cross-controller anti-passing back.

**Request URI Definition****Table B-184 GET /ISAPI/AccessControl/SubmarineBackReader/ConfigureNo/<ID>**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of card readers for cross-controller anti-passing back.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>XML_SubmarineBackReader</u></a> Failed: <a href="#"><u>XML_ResponseStatus</u></a>

**Table B-185 PUT /ISAPI/AccessControl/SubmarineBackReader/ConfigureNo/<ID>**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of card readers for cross-controller anti-passing back.

<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML_SubmarineBackReader</a></u>
<b>Response</b>	<u><a href="#">XML_ResponseStatus</a></u>

**Remarks**

The <ID> in the request URI refers to the configuration No., which is between 1 and 128.

**E.151 /ISAPI/AccessControl/StartReaderInfo**

Operations about first card reader configurations.

**Request URI Definition****Table B-186 GET /ISAPI/AccessControl/StartReaderInfo**

<b>Method</b>	GET
<b>Description</b>	Get the configuration parameters of first card reader.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	<u><a href="#">XML_StartReaderInfo</a></u>

**Table B-187 PUT /ISAPI/AccessControl/StartReaderInfo**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of first card reader.
<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML_StartReaderInfo</a></u>
<b>Response</b>	<u><a href="#">XML_ResponseStatus</a></u>

**E.152 /ISAPI/AccessControl/StartReaderInfo/capabilities**

Get the configuration capability of the first card reader.

## Request URI Definition

**Table B-188 GET /ISAPI/AccessControl/StartReaderInfo/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the first card reader.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_StartReaderInfo</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

## E.153 /ISAPI/AccessControl/SubmarineBackHostInfo/capabilities

Get the configuration capability of access controllers for cross-controller anti-passing back.

## Request URI Definition

**Table B-189 GET /ISAPI/AccessControl/SubmarineBackHostInfo/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of access controllers for cross-controller anti-passing back.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_SubmarineBackHostInfo</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

## E.154 /ISAPI/AccessControl/SubmarineBack/capabilities

Get the configuration capability of the cross-controller anti-passing back server.

## Request URI Definition

**Table B-190 GET /ISAPI/AccessControl/SubmarineBack/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the cross-controller anti-passing back server.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_Cap_SubmarineBack</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

## E.155 /ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID>

Operations about the configuration of access controllers for cross-controller anti-passing back.

### Request URI Definition

**Table B-191 GET /ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID>**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of access controllers for cross-controller anti-passing back.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_SubmarineBackHostInfo</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

**Table B-192 PUT /ISAPI/AccessControl/SubmarineBackHostInfo/ConfigureNo/<ID>**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of access controllers for cross-controller anti-passing back.
<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML_SubmarineBackHostInfo</a></u>
<b>Response</b>	<u><a href="#">XML_ResponseStatus</a></u>

## Remarks

The <ID> in the request URI refers to the configuration No., which is between 1 and 4. More specifically, 1 refers to device No.1 to device No.16, 2 refers to device No.17 to device No.32, 3 refers to device No.33 to device No.48, and 4 refers to device No.49 to device No.64.

## E.156 /ISAPI/AccessControl/SubmarineBack

Operations about the configuration of the cross-controller anti-passing back server.

### Request URI Definition

Table B-193 GET /ISAPI/AccessControl/SubmarineBack

<b>Method</b>	GET
<b>Description</b>	Get the configuration parameters of the cross-controller anti-passing back server.
<b>Query</b>	None.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_SubmarineBack</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

Table B-194 PUT /ISAPI/AccessControl/SubmarineBack

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the cross-controller anti-passing back server.
<b>Query</b>	None.
<b>Request</b>	<u><a href="#">XML_SubmarineBack</a></u>
<b>Response</b>	<u><a href="#">XML_ResponseStatus</a></u>

## E.157 /ISAPI/AccessControl/temperatureMeasureAreaCalibration/capabilities?format=json

Get the calibration configuration capability of the temperature measurement area.

## Request URI Definition

**Table B-195 GET /ISAPI/AccessControl/temperatureMeasureAreaCalibration/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the calibration configuration capability of the temperature measurement area.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_Cap_RegionCalibrationCfg</i> Failed: <i>JSON_ResponseStatus</i>

## Remarks

Calibrating the temperature measurement area can be used to check whether the face position located by the rectangle frame is accurate during thermography.

## E.158 /ISAPI/AccessControl/temperatureMeasureAreaCalibration?format=json

Get or set the calibration parameters of the temperature measurement area.

## Request URI Definition

**Table B-196 GET /ISAPI/AccessControl/temperatureMeasureAreaCalibration?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the calibration parameters of the temperature measurement area.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_RegionCalibrationCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-197 PUT /ISAPI/AccessControl/temperatureMeasureAreaCalibration?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the calibration parameters of the temperature measurement area.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_RegionCalibrationCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.159 /ISAPI/AccessControl/temperatureMeasureAreaCfg/capabilities?format=json

Get the configuration capability of the temperature measurement area.

### Request URI Definition

**Table B-198 GET /ISAPI/AccessControl/temperatureMeasureAreaCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of the temperature measurement area.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_RegionCoordinate</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.160 /ISAPI/AccessControl/temperatureMeasureAreaCfg?format=json

Get or set the parameters of the temperature measurement area.

### Request URI Definition

**Table B-199 GET /ISAPI/AccessControl/temperatureMeasureAreaCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the parameters of the temperature measurement area.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_RegionCoordinate</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-200 PUT /ISAPI/AccessControl/temperatureMeasureAreaCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the parameters of the temperature measurement area.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_RegionCoordinate</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.161 /ISAPI/AccessControl/temperatureMeasureCfg/capabilities?format=json

Get the configuration capability of temperature measurement parameters.

### Request URI Definition

**Table B-201 GET /ISAPI/AccessControl/temperatureMeasureCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the configuration capability of temperature measurement parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_Cap_TemperatureMeasurementCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

## E.162 /ISAPI/AccessControl/temperatureMeasureCfg?format=json

Get or set the temperature measurement parameters.

## Request URI Definition

**Table B-202 GET /ISAPI/AccessControl/temperatureMeasureCfg?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the temperature measurement parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <i>JSON_TemperatureMeasurementCfg</i> Failed: <i>JSON_ResponseStatus</i>

**Table B-203 PUT /ISAPI/AccessControl/temperatureMeasureCfg?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the temperature measurement parameters.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<i>JSON_TemperatureMeasurementCfg</i>
<b>Response</b>	<i>JSON_ResponseStatus</i>

## E.163 /ISAPI/AccessControl/UserInfo/capabilities?format=json

Get the person management capability.

## Request URI Definition

**Table B-204 GET /ISAPI/AccessControl/UserInfo/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the person management capability.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.

Request	None.
Response	Succeeded: <u>JSON_Cap_UserInfo</u> Failed: <u>JSON_ResponseStatus</u>

## E.164 /ISAPI/AccessControl/UserInfo/Count?format=json

Get the total number of the added persons.

### Request URI Definition

**Table B-205 GET /ISAPI/AccessControl/UserInfo/Count?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the total number of the added persons.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u>JSON_UserInfoCount</u> Failed: <u>JSON_ResponseStatus</u>

### Remarks

This URI is not supported by integration of information release system.

## E.165 /ISAPI/AccessControl/UserInfo/Delete?format=json

Delete person information only.

### Request URI Definition

**Table B-206 PUT /ISAPI/AccessControl/UserInfo/Delete?format=json**

<b>Method</b>	PUT
<b>Description</b>	Delete person information only.

<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_UserInfoDelCond</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**E.166 /ISAPI/AccessControl/UserInfo/Modify?format=json**

Edit person information.

**Request URI Definition****Table B-207 PUT /ISAPI/AccessControl/UserInfo/Modify?format=json**

<b>Method</b>	PUT
<b>Description</b>	Edit person information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_UserInfo</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**E.167 /ISAPI/AccessControl/UserInfo/Record?format=json**

Add a person.

**Request URI Definition****Table B-208 POST /ISAPI/AccessControl/UserInfo/Record?format=json**

<b>Method</b>	POST
<b>Description</b>	Add a person.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_UserInfo</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**E.168 /ISAPI/AccessControl/UserInfo/Search?format=json**

Search for person information.

## Request URI Definition

**Table B-209 POST /ISAPI/AccessControl/UserInfo/Search?format=json**

<b>Method</b>	POST
<b>Description</b>	Search for person information.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	<u><a href="#">JSON_UserInfoSearchCond</a></u>
<b>Response</b>	<u><a href="#">JSON_UserInfoSearch</a></u>

## Remarks

The Request (user information search condition [JSON\\_UserInfoSearchCond](#)) depends on the user information capability JSON\_Cap\_UserInfo (related node: <UserInfoSearchCond>).

## E.169 /ISAPI/AccessControl/UserInfo/SetUp?format=json

Set person information.

## Request URI Definition

**Table B-210 PUT /ISAPI/AccessControl/UserInfo/SetUp?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set person information.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_UserInfo</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

- If the device has checked that the person does not exist according to the employee No. (person ID), the person information will be added.
- If the device has checked that the person already exists according to the employee No. (person ID), the person information will be edited.
- If a person needs to be deleted, the **deleteUser** in the message **JSON\_UserInfo** should be set to "true", and the success response message will be returned no matter whether the person information exists or not. Deleting the person will only delete the person's information and will not delete the linked cards, fingerprints, and face information.

## E.170 /ISAPI/AccessControl/UserInfoDetail/Delete/capabilities?format=json

Get the capability of deleting person information (including linked cards, fingerprints, and faces) and permissions.

### Request URI Definition

Table B-211 GET /ISAPI/AccessControl/UserInfoDetail/Delete/capabilities?format=json

<b>Method</b>	GET
<b>Description</b>	Get the capability of deleting person information (including linked cards, fingerprints, and faces) and permissions.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <b><u>JSON_Cap_UserInfoDetail</u></b> Failed: <b><u>JSON_ResponseStatus</u></b>

## Remarks

This URI is not supported by integration of information release system.

## E.171 /ISAPI/AccessControl/UserInfoDetail/Delete?format=json

Start deleting all person information and permissions by employee No.

### Request URI Definition

**Table B-212 PUT /ISAPI/AccessControl/UserInfoDetail/Delete?format=json**

<b>Method</b>	PUT
<b>Description</b>	Start deleting all person information (including linked cards, fingerprints, and faces) and permissions by employee No.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON.UserInfoDetail</a></u>
<b>Response</b>	<u><a href="#">JSON.ResponseStatus</a></u>

### Remarks

- This URI is only used to start deleting. To check whether the deleting is completed, you should call the request URI **/ISAPI/AccessControl/UserInfoDetail/DeleteProcess?format=json** by GET method to get the deleting status.
- This URI is not supported by integration of information release system.

## E.172 /ISAPI/AccessControl/UserInfoDetail/DeleteProcess?format=json

Get the status of deleting all person information and permissions by employee No.

### Request URI Definition

**Table B-213 GET /ISAPI/AccessControl/UserInfoDetail/DeleteProcess?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the status of deleting all person information (including linked cards, fingerprints, and faces) and permissions by employee No.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.

<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON.UserInfoDetailDeleteProcess</u></a> Failed: <a href="#"><u>JSONResponseStatus</u></a>

**Remarks**

- When starting deleting all person information (including linked cards, fingerprints, and faces) and permissions by employee No., this URI will be repeatedly called to get the deleting status until "success" or "failed" is returned by the parameter **status** in the message [JSON.UserInfoDetailDeleteProcess](#).
- This URI is not supported by integration of information release system.

**E.173 /ISAPI/AccessControl/UserRightPlanTemplate/capabilities?format=json**

Get the schedule template configuration capability of the access permission control.

**Request URI Definition**

**Table B-214 GET /ISAPI/AccessControl/UserRightPlanTemplate/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the schedule template configuration capability of the access permission control.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON.Cap_UserRightPlanTemplate</u></a> Failed: <a href="#"><u>JSONResponseStatus</u></a>

**Remarks**

This URI is not supported by integration of information release system.

## E.174 /ISAPI/AccessControl/UserRightPlanTemplate/<TemplateNo>?format=json

Operations about the schedule template configuration of the access permission control.

### Request URI Definition

**Table B-215 GET /ISAPI/AccessControl/UserRightPlanTemplate/<TemplateNo>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the schedule template configuration parameters of the access permission control.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No. <b>&lt;TemplateNo&gt;:</b> int, schedule template ID, start from 1.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_UserRightPlanTemplate</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-216 PUT /ISAPI/AccessControl/UserRightPlanTemplate/<TemplateNo>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the schedule template parameters of the access permission control.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>&lt;TemplateNo&gt;:</b> int, schedule template ID, start from 1.
<b>Request</b>	<u><a href="#">JSON_UserRightPlanTemplate</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

### Remarks

- The **<TemplateNo>** in the request URI refers to the schedule template No. which starts from 1, and you can get the maximum number of the templates supported by the device from the

- schedule template configuration capability of the access permission control ([JSON\\_Cap\\_UserRightPlanTemplate](#)).  
  - This URI is not supported by integration of information release system.

## E.175 /ISAPI/AccessControl/UserRightHolidayGroupCfg/<GroupNo>?format=json

Operations about the holiday group configuration of the access permission control schedule.

### Request URI Definition

Table B-217 GET /ISAPI/AccessControl/UserRightHolidayGroupCfg/<GroupNo>?format=json

<b>Method</b>	GET
<b>Description</b>	Get the holiday group configuration parameters of the access permission control schedule.
<b>Query</b>	<b>format</b> : determine the format of request or response message. <b>terminalNo</b> : dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No. <b>&lt;GroupNo&gt;</b> : int, holiday group ID, start from 1.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_UserRightHolidayGroupCfg</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

Table B-218 PUT /ISAPI/AccessControl/UserRightHolidayGroupCfg/<GroupNo>?format=json

<b>Method</b>	PUT
<b>Description</b>	Set the holiday group parameters of the access permission control schedule.
<b>Query</b>	<b>format</b> : determine the format of request or response message. <b>&lt;GroupNo&gt;</b> : int, holiday group ID, start from 1.
<b>Request</b>	<u><a href="#">JSON_UserRightHolidayGroupCfg</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

- The <GroupNo> in the request URI refers to the holiday group No. which starts from 1, and you can get the maximum number of the holiday groups supported by the device from the holiday group configuration capability of the access permission control schedule ([JSON\\_Cap\\_UserRightHolidayGroupCfg](#)).
- This URI is not supported by integration of information release system.

## E.176 /ISAPI/AccessControl/UserRightHolidayGroupCfg/capabilities?format=json

Get the holiday group configuration capability of the access permission control schedule.

### Request URI Definition

**Table B-219 GET /ISAPI/AccessControl/UserRightHolidayGroupCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the holiday group configuration capability of the access permission control schedule.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_Cap_UserRightHolidayGroupCfg</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

## Remarks

This URI is not supported by integration of information release system.

## E.177 /ISAPI/AccessControl/UserRightHolidayPlanCfg/<PlanNo>?format=json

Operations about the holiday schedule configuration of the access permission control.

## Request URI Definition

**Table B-220 GET /ISAPI/AccessControl/UserRightHolidayPlanCfg/<PlanNo>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the holiday schedule configuration parameters of the access permission control.
<b>Query</b>	<p><b>format:</b> determine the format of request or response message.</p> <p><b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.</p> <p>&lt;<b>PlanNo</b>&gt;: int, holiday schedule ID.</p>
<b>Request</b>	None.
<b>Response</b>	<p>Succeeded: <a href="#"><u>JSON_UserRightHolidayPlanCfg</u></a></p> <p>Failed: <a href="#"><u>JSON_ResponseStatus</u></a></p>

**Table B-221 PUT /ISAPI/AccessControl/UserRightHolidayPlanCfg/<PlanNo>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the holiday schedule parameters of the access permission control.
<b>Query</b>	<p><b>format:</b> determine the format of request or response message.</p> <p>&lt;<b>PlanNo</b>&gt;: int, holiday schedule ID.</p>
<b>Request</b>	<a href="#"><u>JSON_UserRightHolidayPlanCfg</u></a>
<b>Response</b>	<a href="#"><u>JSON_ResponseStatus</u></a>

## Remarks

- The <**PlanNo**> in the request URI refers to the holiday schedule No. which starts from 1, and you can get the maximum number of the holiday schedules supported by the device from the holiday schedule configuration capability of the access permission control ([JSON\\_Cap\\_UserRightHolidayPlanCfg](#)).
- This URI is not supported by integration of information release system.

## E.178 /ISAPI/AccessControl/UserRightHolidayPlanCfg/capabilities?format=json

Get the holiday schedule configuration capability of the access permission control.

### Request URI Definition

**Table B-222 GET /ISAPI/AccessControl/UserRightHolidayPlanCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the holiday schedule configuration capability of the access permission control.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <b><i>JSON_Cap_UserRightHolidayPlanCfg</i></b> Failed: <b><i>JSON_ResponseStatus</i></b>

### Remarks

This URI is not supported by integration of information release system.

## E.179 /ISAPI/AccessControl/UserRightWeekPlanCfg/<PlanNo>?format=json

Operations about the week schedule configuration of the access permission control.

### Request URI Definition

**Table B-223 GET /ISAPI/AccessControl/UserRightWeekPlanCfg/<PlanNo>?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the week schedule configuration parameters of the access permission control.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

	<p><b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.</p> <p><b>&lt;PlanNo&gt;:</b> int, week schedule ID.</p>
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON UserRightWeekPlanCfg</a></u> Failed: <u><a href="#">JSON ResponseStatus</a></u>

**Table B-224 PUT /ISAPI/AccessControl/UserRightWeekPlanCfg/<PlanNo>?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the week schedule parameters of the access permission control.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>&lt;PlanNo&gt;:</b> int, week schedule ID.
<b>Request</b>	<u><a href="#">JSON UserRightWeekPlanCfg</a></u>
<b>Response</b>	<u><a href="#">JSON ResponseStatus</a></u>

**Remarks**

- The **<PlanNo>** in the request URI refers to the week schedule No. which starts from 1, and you can get the maximum number of the week schedules supported by the device from the week schedule configuration capability of the access permission control ([JSON Cap UserRightWeekPlanCfg](#)).
- This URI is not supported by integration of information release system.

**E.180 /ISAPI/AccessControl/UserRightWeekPlanCfg/capabilities?format=json**

Get the week schedule configuration capability of the access permission control.

## Request URI Definition

**Table B-225 GET /ISAPI/AccessControl/UserRightWeekPlanCfg/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the week schedule configuration capability of the access permission control.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_Cap_UserRightWeekPlanCfg</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

## Remarks

This URI is not supported by integration of information release system.

## E.181 /ISAPI/Intelligent/FDLib/capabilities?format=json

Get face picture library capability.

## Request URI Definition

**Table B-226 GET /ISAPI/Intelligent/FDLib/capabilities?format=json**

<b>Method</b>	GET
<b>Description</b>	Get face picture library capability.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> int, terminal No., starts from 1.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_FPLibCap</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

**E.182 /ISAPI/Intelligent/FDLib?format=json**

Operations about the face picture library.

**Request URI Definition****Table B-227 POST /ISAPI/Intelligent/FDLib?format=json**

<b>Method</b>	POST
<b>Description</b>	Create a face picture library
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	<b><u>JSON_CreateFPLibCond</u></b>
<b>Response</b>	Succeeded: <b><u>JSON_CreateFPLibResult</u></b> Failed: <b><u>JSON_ResponseStatus</u></b>

**Table B-228 GET /ISAPI/Intelligent/FDLib?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the information, including library ID, library type, name, and custom information, of all face picture libraries.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None
<b>Response</b>	Succeeded: <b><u>JSON_FPLibListInfo</u></b> Failed: <b><u>JSON_ResponseStatus</u></b>

**Table B-229 DELETE /ISAPI/Intelligent/FDLib?format=json**

<b>Method</b>	DELETE
<b>Description</b>	Delete all face picture libraries.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None
<b>Response</b>	<b><i>JSONResponseStatus</i></b>

**Remarks**

- After a face picture library is created, the face picture library ID will be returned. Each face picture library ID of the same library type is unique.
- The POST and DELETE operation methods are not supported by integration of information release system.

**E.183 /ISAPI/Intelligent/FDLib?format=json&FDID=&faceLibType=**

Operations about the management of a specific face picture library.

**Request URI Definition****Table B-230 GET /ISAPI/Intelligent/FDLib?format=json&FDID=&faceLibType=**

<b>Method</b>	GET
<b>Description</b>	Get the information, including library ID, library type, name, and custom information, of a specific face picture library.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>FDID:</b> optional, string, face picture library ID. <b>faceLibType:</b> optional, string, face picture library type, which can be "blackFD" (list library) or "staticFD" (static library). <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management

	server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <a href="#"><u>JSON_SingleFPLibInfo</u></a> Failed: <a href="#"><u>JSON_ResponseStatus</u></a>

**Table B-231 PUT /ISAPI/Intelligent/FDLib?format=json&FDID=&faceLibType=**

<b>Method</b>	PUT
<b>Description</b>	Edit the information of a specific face picture library information, including name and custom information.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>FDID:</b> optional, string, face picture library ID <b>faceLibType:</b> optional, string, face picture library type, which can be "blackFD" (list library) or "staticFD" (static library). <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	<a href="#"><u>JSON_EditFPLibInfo</u></a>
<b>Response</b>	<a href="#"><u>JSON_ResponseStatus</u></a>

**Table B-232 DELETE /ISAPI/Intelligent/FDLib?format=json&FDID=&faceLibType=**

<b>Method</b>	DELETE
<b>Description</b>	Delete a specific face picture library.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>FDID:</b> face picture library ID <b>faceLibType:</b> face picture library type <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.

<b>Request</b>	None.
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

**Remarks**

- In the URI, to specify a face picture library, both the library ID (**FDID**) and library type (**faceLibType**) are required.
- This URI is not supported by integration of information release system.

**E.184 /ISAPI/Intelligent/FDLib/Count?format=json**

Get the total number of face records in all face picture libraries.

**Request URI Definition**

**Table B-233 GET /ISAPI/Intelligent/FDLib/Count?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the total number of face records in all face picture libraries.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None
<b>Response</b>	Succeeded: <u><a href="#">JSON_FaceRecordNumInAllFPLib</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Remarks**

This URI is not supported by integration of information release system.

**E.185 /ISAPI/Intelligent/FDLib/Count?  
format=json&FDID=&faceLibType=**

Get the number of face records in a specific face picture library.

## Request URI Definition

**Table B-234 GET /ISAPI/Intelligent/FDLib/Count?format=json&FDID=&faceLibType=**

<b>Method</b>	GET
<b>Description</b>	Get the number of face records in a specific face picture library.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>FDID:</b> face picture library ID. <b>faceLibType:</b> face picture library type, which can equal to "blackFD" (list library) and "staticFD" (static library). <b>terminalNo:</b> dependent, int, terminal No., starts from 1. It is required for information release system integration. The No. is generated after information release terminal is registered to the central management server, and you can call "/ISAPI/Publish/TerminalMgr/terminalSearch" by POST method to get the generated terminal No.
<b>Request</b>	None
<b>Response</b>	Succeeded: <b><u>JSON_FaceRecordNumInOneFPLib</u></b> Failed: <b><u>JSON_ResponseStatus</u></b>

## Remarks

- In the URI, to specify a face picture library, both the library ID (**FDID**) and library type (**faceLibType**) are required, e.g., /ISAPI/Intelligent/FDLib/Count?format=json&FDID=122334455566788&faceLibType=blackFD.
- This URI is not supported by integration of information release system.

## E.186 /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json

Add the face record (face picture and person information) to a face picture library or multiple face picture libraries.

## Request URI Definition

**Table B-235 POST /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json**

<b>Method</b>	POST
<b>Description</b>	Add a face record (including face picture and person information) to the face picture library.
<b>Query</b>	<b>format:</b> determine the format of request or response message.

<b>Request</b>	<u><a href="#">JSON_AddFaceRecordCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_AddFaceRecordResult</a></u> Failed: <u><a href="#">JSONResponseStatus</a></u>

**Remarks**

- The face picture in the face record supports URL and binary data format. If the JSON message about adding condition parameters ([JSON\\_AddFaceRecordCond](#)) contains node "faceURL", the picture should be uploaded in URL format; otherwise, the picture should be uploaded in binary data format.
- Currently, the picture URL refers to the URL generated when the picture is stored in the storage server of central management server. Before using picture URL, you should generate the URL via the cloud storage server of storage service component of ISUP (Intelligent Security Uplink Protocol).

**E.187 /ISAPI/Intelligent/FDLib/FDSearch?format=json**

Search for the face records in the face picture library.

**Request URI Definition**

**Table B-236 POST /ISAPI/Intelligent/FDLib/FDSearch?format=json**

<b>Method</b>	POST
<b>Description</b>	Search for the face records in the a face picture library or multiple face picture libraries. Fuzzy search is also supported.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>terminalNo:</b> int, terminal No., starts from 1.
<b>Request</b>	<u><a href="#">JSON_SearchFaceRecordCond</a></u>
<b>Response</b>	Succeeded: <u><a href="#">JSON_SearchFaceRecordResult</a></u> Failed: <u><a href="#">JSONResponseStatus</a></u>

**E.188 /ISAPI/Intelligent/FDLib/FDModify?format=json**

Edit face records in the face picture library in a batch.

## Request URI Definition

**Table B-237 PUT /ISAPI/Intelligent/FDLib/FDModify?format=json**

<b>Method</b>	PUT
<b>Description</b>	Edit face records in the face picture library in a batch.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_BatchEditFaceRecord</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## E.189 /ISAPI/Intelligent/FDLib/FDSearch/Delete? format=json&FDID=&faceLibType=

Delete the face record(s) in a specific face picture library.

## Request URI Definition

**Table B-238 PUT /ISAPI/Intelligent/FDLib/FDSearch/Delete?format=json&FDID=&faceLibType=**

<b>Method</b>	PUT
<b>Description</b>	Delete the face record(s) in the face picture library.
<b>Query</b>	<b>format:</b> determine the format of request or response message. <b>FDID:</b> face picture library ID <b>faceLibType:</b> face picture library type
<b>Request</b>	<u><a href="#">JSON_DelFaceRecord</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

In the URI, to specify a face picture library, both the library ID (**FDID**) and library type (**faceLibType**) are required.

## E.190 /ISAPI/Intelligent/FDLib/FDSetUp?format=json

Set the face record (including face picture, person information, etc.) in the face picture library.

## Request URI Definition

**Table B-239 PUT /ISAPI/Intelligent/FDLib/FDSetUp?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the face record (including face picture, person information, etc.) in the face picture library.
<b>Query</b>	<b>format:</b> determine the format of request or response message.
<b>Request</b>	<u><a href="#">JSON_SetFaceRecord</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

## Remarks

- If the face picture with the employee No. (person ID) does not exist, the face record will be added.
- If the face picture with the employee No. (person ID) exists, the face record will be overwritten.
- When deleting the face record, the **faceLibType**, **FDID**, **FPID**, and **deleteFP** in the request message [JSON\\_SetFaceRecord](#) should be configured, and the success response message will be returned no matter whether deleting succeeded or not.
- The employee No. is required.

## E.191 /ISAPI/System/capabilities

Get device capability.

## Request URI Definition

**Table B-240 GET /ISAPI/System/capabilities**

<b>Method</b>	GET
<b>Description</b>	Get device capability.
<b>Query</b>	None
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">XML_DeviceCap</a></u> Failed: <u><a href="#">XML_ResponseStatus</a></u>

## E.192 /ISAPI/System/PictureServer?format=json

Operations about the picture storage server configuration parameters.

### Request URI Definition

**Table B-241 GET /ISAPI/System/PictureServer?format=json**

<b>Method</b>	GET
<b>Description</b>	Get the picture storage server parameters.
<b>Query</b>	<b>format</b> : determine the format of request or response message. <b>security</b> : the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. <b>iv</b> : the initialization vector, and it is required when <b>security</b> is 1 or 2.
<b>Request</b>	None.
<b>Response</b>	Succeeded: <u><a href="#">JSON_PictureServerInformation</a></u> Failed: <u><a href="#">JSON_ResponseStatus</a></u>

**Table B-242 PUT /ISAPI/System/PictureServer?format=json**

<b>Method</b>	PUT
<b>Description</b>	Set the picture storage server parameters.
<b>Query</b>	<b>format</b> : determine the format of request or response message. <b>security</b> : the version No. of encryption scheme. When <b>security</b> does not exist, it indicates that the data is not encrypted; when <b>security</b> is 1, it indicates that the nodes of sensitive information in the message are encrypted in AES128 CBC mode; when <b>security</b> is 2, it indicates that the nodes of sensitive information in the message are encrypted in AES256 CBC mode. <b>iv</b> : the initialization vector, and it is required when <b>security</b> is 1 or 2.
<b>Request</b>	<u><a href="#">JSON_PictureServerInformation</a></u>
<b>Response</b>	<u><a href="#">JSON_ResponseStatus</a></u>

# Appendix F. Request and Response Messages

## F.1 JSON\_AcsCfg

AcsCfg message in JSON format

```
{
  "AcsCfg": {
    "RS485Backup": ,
    /*optional, boolean, whether to enable downstream RS-485 communication redundancy: "true"-yes, "false"-no*/
    "showCapPic": ,
    /*optional, boolean, whether to display the captured picture: "true"-yes, "false"-no*/
    "showUserInfo": ,
    /*optional, boolean, whether to display user information: "true"-yes, "false"-no*/
    "overlayUserInfo": ,
    /*optional, boolean, whether to overlay user information: "true"-yes, "false"-no*/
    "voicePrompt": ,
    /*optional, boolean, whether to enable audio announcement: "true"-yes, "false"-no*/
    "uploadCapPic": ,
    /*optional, boolean, whether to upload the picture from linked capture: "true"-yes, "false"-no*/
    "saveCapPic": ,
    /*optional, boolean, whether to save the captured picture: "true"-yes, "false"-no*/
    "inputCardNo": ,
    /*optional, boolean, whether to allow inputting card No. on keypad: "true"-yes, "false"-no*/
    "enableWifiDetect": ,
    /*optional, boolean, whether to enable Wi-Fi probe: "true"-yes, "false"-no*/
    "enable3G4G": ,
    /*optional, boolean, whether to enable 3G/4G: "true"-yes, "false"-no*/
    "protocol": "",
    /*optional, string, communication protocol type of the card reader: "Private"-private protocol, "OSDP"-OSDP protocol*/
    "enableCaptureCertificate": true,
    /*optional, boolean, whether to enable capturing the ID picture: true (yes), false (no). The captured ID picture will be compared with the captured face picture to check whether it is the same person. If this node does not exist, it indicates that this function is not supported*/
    "showPicture": ,
    /*optional, boolean, whether to display the authenticated picture: true-display, false-not display*/
    "showEmployeeNo": ,
    /*optional, boolean, whether to display the authenticated employee ID: true-
```

```
display, false-not display*/
    "showName": ,
/*optional, boolean, whether to display the authenticated name: true-display,
false-not display*/
    "desensitiseEmployeeNo": ,
/*dependent, boolean, whether to enable employee No. de-identification for
local UI display: true (yes), false (no). This node is valid when the value of
the node showEmployeeNo is true*/
    "desensitiseName": ,
/*dependent, boolean, whether to enable name de-identification for local UI
display: true (yes), false (no). This node is valid when the value of the node
showName is true*/
    "thermalEnabled": true,
/*optional, boolean, whether to enable temperature measurement: true-enable
(default), false-disable*/
    "thermalMode": true,
/*optional, boolean, whether to enable temperature measurement only mode: true-
enable (only for temperature measurement), false-disable (default)*/
    "thermalPictureEnabled": true,
/*optional, boolean, whether to enable uploading visible light pictures in
temperature measurement only mode: true-enable, false-disable (default). This
field is used to control uploading captured pictures and visible light
pictures*/
    "thermalIp": "192.168.1.1",
/*optional, string, IP address of the thermography device. For access control
devices, each device only requires one IP address; for metal detector doors,
this field does not need to be configured*/
    "highestThermalThreshold": 37.3,
/*optional, float, upper limit of the temperature threshold which is accurate
to one decimal place, unit: Celsius. The default value of this field is
37.3 °C. This field is used to check whether to open the door when the
temperature is above the upper limit*/
    "lowestThermalThreshold": ,
/*optional, float, lower limit of the temperature threshold which is accurate
to one decimal place, unit: Celsius. This field is used to check whether to
open the door when the temperature is below the lower limit*/
    "thermalDoorEnabled": false,
/*optional, boolean, whether to open the door when the temperature is above the
upper limit (highestThermalThreshold) or below the lower limit
(lowestThermalThreshold) of the threshold: true-open the door, false-not open
the door (default)*/
    "QRCodeEnabled": false,
/*optional, boolean, whether to enable QR code function: true-enable, false-
disable (default)*/
    "remoteCheckDoorEnabled": false,
/*optional, boolean, whether to enable controlling the door by remote
verification: true-control, false-not control (default)*/
    "checkChannelType": "",
/*dependent, string, verification channel type: "Ezviz"-EZVIZ channel, "ISUP"-_
ISUP channel, "ISAPI"-ISAPI channel, "PrivateSDK"-private SDK channel,
"ISAPIListen"-ISAPI listening channel. This field is valid when
remoteCheckDoorEnabled is true*/
```

```

    "channelIp": "",
/*dependent, string, IP address of the verification channel. This field is
valid when checkChannelType is "PrivateSDK"*/
    "uploadVerificationPic": ,
/*optional, boolean, whether to upload the authenticated picture: true, false*/
    "saveVerificationPic": ,
/*optional, boolean, whether to save the authenticated picture: true, false*/
    "saveFacePic": ,
/*optional, boolean, whether to save the registered face picture: true, false*/
    "thermalUnit": "",
/*optional, string, temperature unit: "celsius" (default), "fahrenheit". If
this node does not exist, the default unit is Celsius*/
    "highestThermalThresholdF": ,
/*optional, float, the maximum value of the temperature threshold. The value is
accurate to one decimal place, and the unit is Fahrenheit. This node is used to
check whether to open the door when the temperature is higher than the
threshold*/
    "lowestThermalThresholdF": ,
/*optional, float, the minimum value of the temperature threshold. The value is
accurate to one decimal place, and the unit is Fahrenheit. This node is used to
check whether to open the door when the temperature is lower than the
threshold*/
    "thermalCompensation": ,
/*optional, float, temperature compensation, the value is accurate to one
decimal place. The unit depends on the node thermalUnit. If the node
thermalUnit does not exist, the default unit is Celsius*/
}
}

```

## F.2 JSON\_AcsEventCond

AcsEventCond message in JSON format

```

{
    "AcsEventCond": {
        "searchID": "",
/*required, string type, search ID, which is used to confirm the upper-level
platform or system. If the platform or the system is the same one during two
searching, the search history will be saved in the memory to speed up next
searching*/
        "searchResultPosition": ,
/*required, integer, the start position of the search result in the result
list. When there are multiple records and you cannot get all search results at
a time, you can search for the records after the specified position next time*/
        "maxResults": ,
/*required, integer, maximum number of search results. If maxResults exceeds
the range returned by the device capability, the device will return the maximum
number of search results according to the device capability and will not return
error message*/
        "major": ,
    }
}

```

```

/*required, integer, major alarm/event types (the type value should be
transformed to the decimal number), see Access Control Alarm Types for details*/
    "minor": ,
/*required, integer, minor alarm/event types (the type value should be
transformed to the decimal number), see Access Control Alarm Types for details*/
    "startTime": "",
/*optional, string, start time (UTC time), e.g., 2016-12-12T17:30:08+08:00*/
    "endTime": "",
/*optional, string, end time (UTC time), e.g., 2017-12-12T17:30:08+08:00*/
    "cardNo": "",
/*optional, string, card No.*/
    "name": "",
/*optional, string, cardholder name*/
    "picEnable": ,
/*optional, boolean, whether to contain pictures: "false"-no, "true"-yes*/
    "beginSerialNo": ,
/*optional, integer, start serial No.*/
    "endSerialNo": ,
/*optional, integer, end serial No.*/
    "employeeNoString": "",
/*optional, string, employee No. (person ID)*/
    "eventAttribute": "",
/*optional, string, event attribute: "attendance"-valid authentication,
"other"*/
    "employeeNo": "",
/*optional, string, employee No. (person ID)*/
    "timeReverseOrder": ,
/*optional, boolean, whether to return events in descending order of time
(later events will be returned first): true-yes, false or this node is not
returned-no*/
    "isAbnormalTemperature":true
/*optional, boolean, whether the skin-surface temperature is abnormal*/
    "temperatureSearchCond": "all"
/*optional, string, temperature search conditions, all (events with temperature
information), normal (events with normal temperature), abnormal (events with
abnormal temperature), if it exists with isAbnormalTemperature, then the latter
will be invalid.*/
}
}

```

### F.3 JSON\_AcsEvent

AcsEvent message in JSON format

```

{
    "AcsEvent":{
        "searchID": "",
/*required, string type, search ID, which is used to confirm the upper-level
platform or system. If the platform or the system is the same one during two
searching, the search history will be saved in the memory to speed up next
}
}

```

```
searching*/
    "responseStatusStrg": "",
/*required, string, search status: "OK"-searching completed, "MORE"-searching
for more results, "NO MATCH"-no matched results*/
    "numOfMatches": ,
/*required, integer, number of returned results*/
    "totalMatches": ,
/*required, integer, total number of matched results*/
    "InfoList": [],
/*optional, event details*/
    "major": ,
/*required, integer, major alarm/event types (the type value should be
transformed to the decimal number), see Access Control Event Types for details*/
    "minor": ,
/*required, integer, minor alarm/event types (the type value should be
transformed to the decimal number), see Access Control Event Types for details*/
    "time": "",
/*required, string, time (UTC time), e.g., "2016-12-12T17:30:08+08:00"*/
    "netUser": "",
/*optional, string, user name*/
    "remoteHostAddr": "",
/*optional, string, remote host address*/
    "cardNo": "",
/*optional, string, card No.*/
    "cardType": ,
/*optional, integer, card types: 1-normal card, 2-disabled card, 3-blocklist
card, 4-patrol card, 5-duress card, 6-super card, 7-visitor card, 8-dismiss
card*/
    "name": "",
/*optional, string, person name*/
    "whiteListNo": ,
/*optional, integer, allowlist No., which is between 1 and 8*/
    "reportChannel": ,
/*optional, integer, channel type for uploading alarm/event: 1-for uploading
arming information, 2-for uploading by central group 1, 3-for uploading by
central group 2*/
    "cardReaderKind": ,
/*optional, integer, authentication unit type: 1-IC card reader, 2-ID card
reader, 3-QR code scanner, 4-fingerprint module*/
    "cardReaderNo": ,
/*Optional, integer, authentication unit No.*/
    "doorNo": ,
/*optional, integer, door or floor No.*/
    "verifyNo": ,
/*optional, integer, multiple authentication No.*/
    "alarmInNo": ,
/*optional, integer, alarm input No.*/
    "alarmOutNo": ,
/*optional, integer, alarm output No.*/
    "caseSensorNo": ,
/*optional, integer, event trigger No.*/
    "RS485No": ,
```

```
/*optional, integer, RS-485 channel No.*/
    "multiCardGroupNo": ,
/*optional, integer, group No.*/
    "accessChannel": ,
/*optional, integer, swing barrier No.*/
    "deviceNo": ,
/*optional, integer, device No.*/
    "distractControlNo": ,
/*optional, integer, distributed controller No.*/
    "employeeNoString": "",
/*optional, integer, employee No. (person ID)*/
    "localControllerID": ,
/*optional, integer, distributed access controller No.: 0-access controller, 1
to 64-distributed access controller No.1 to distributed access controller No.
64*/
    "InternetAccess": ,
/*optional, integer, network interface No.: 1-upstream network interface No.1,
2-upstream network interface No.2, 3-downstream network interface No.1*/
    "type": ,
/*optional, integer, zone type: 0-instant alarm zone, 1-24-hour alarm zone, 2-
delayed zone, 3-internal zone, 4-key zone, 5-fire alarm zone, 6-perimeter
protection, 7-24-hour silent alarm zone, 8-24-hour auxiliary zone, 9-24-hour
shock alarm zone, 10-emergency door open alarm zone, 11-emergency door closed
alarm zone, 255-none*/
    "MACAddr": "",
/*optional, string, physical address*/
    "swipeCardType": ,
/*optional, integer, card swiping types: 0-invalid, 1-QR code*/
    "serialNo": ,
/*optional, integer, event serial No., which is used to judge whether the event
loss occurred*/
    "channelControllerID": ,
/*optional, integer, lane controller No.: 1-main lane controller, 2-sub lane
controller*/
    "channelControllerLampID": ,
/*optional, integer, light board No. of lane controller, which is between 1 and
255*/
    "channelControllerIRAdaptorID": ,
/*optional, integer, IR adapter No. of lane controller, which is between 1 and
255*/
    "channelControllerIREmitterID": ,
/*optional, integer, active infrared intrusion detector No. of lane controller,
which is between 1 and 255*/
    "userType": "",
/*optional, string, person type: "normal"-normal person (household), "visitor"-visitor,
"blacklist"-person in blocklist, "administrators"-administrator*/
    "currentVerifyMode": "",
/*optional, string, authentication mode: "cardAndPw"-card+password, "card",
"cardOrPw"-card or password, "fp"-fingerprint, "fpAndPw"-fingerprint+password,
"fpOrCard"-fingerprint or card, "fpAndCard"-fingerprint+card, "fpAndCardAndPw"-fingerprint+card+password,
"faceOrFpOrCardOrPw"-face or fingerprint or card or password, "faceAndFp"-face+fingerprint,
"faceAndPw"-face+password,
```

```

"faceAndCard"-face+card, "face", "employeeNoAndPw"-employee No.+password,
"fpOrPw"-fingerprint or password, "employeeNoAndFp"-employee No.+fingerprint,
"employeeNoAndFpAndPw"-employee No.+fingerprint+password, "faceAndFpAndCard"-  

face+fingerprint+card, "faceAndPwAndFp"-face+password+fingerprint,  

"employeeNoAndFace"-employee No.+face, "faceOrfaceAndCard"-face or face+card,  

"fpOrface"-fingerprint or face, "cardOrfaceOrPw"-card or face or password,  

"cardOrFpOrPw"-card or fingerprint or password*
    "QRCodeInfo":"test",
/*optional, string, QR code information*/
    "thermometryUnit": "",  

/*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheit"-  

Fahrenheit, "kelvin"-Kelvin*/
    "currTemperature": ,
/*optional, float, face temperature which is accurate to one decimal place*/
    "isAbnormalTemperature": ,
/*optional, boolean, whether the face temperature is abnormal: true-yes, false-  

no*/
    "RegionCoordinates":{
/*optional, face temperature's coordinates*/
        "positionX": ,
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
        "positionY": ,
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
    },
    "mask": "",  

/*optional, string, whether the person is wearing mask: "unknown", "yes"-  

wearing mask, "no"-not wearing mask*/
    "pictureURL": "",  

/*optional, string, picture URL*/
    "filename": "",  

/*optional, string, file name. If multiple pictures are returned at a time,  

filename of each picture should be unique*/
    "attendanceStatus": "",  

/*optional, string, attendance status: "undefined", "checkIn"-check in,  

"checkOut"-check out, "breakOut"-break out, "breakIn"-break in, "overtimeIn"-  

overtime in, "overTimeOut"-overtime out*/
    "label": "",  

/*optional, string, custom attendance name*/
    "statusValue": ,  

/*optional, integer, status value*/
    "helmet": "",  

/*optional, string, whether the person is wearing hard hat: "unknown", "yes"-  

wearing hard hat, "no"-not wearing hard hat*/
    "visibleLightPicUrl": "test",
/*optional, string, URL of the visible light picture*/
    "thermalPicUrl": "test",
/*optional, string, URL of the thermal picture*/
    "appType": "attendance",
/*optional, string, application type: "attendance" (attendance application),  

"signIn" (check-in application, which is only used for information release  

products)*/
    "HealthInfo": {

```

```
/*optional, object, health information*/
    "healthCode": 1,
/*optional, int, health code status: 0 (no request), 1 (no health code), 2
(green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6
(other error, e.g., searching failed due to API exception), 7 (searching for
the health code timed out)*/
    "NADCode": 1,
/*optional, int, nucleic acid test result: 0 (no result), 1 (negative, which
means normal), 2 (positive, which means diagnosed), 3 (the result has expired)*/
    "travelCode": 1,
/*optional, int, trip code: 0 (no trip in the past 14 days), 1 (once left in
the past 14 days), 2 (has been to the high-risk area in the past 14 days), 3
(other)*/
    "vaccineStatus": 1
/*optional, int, whether the person is vaccinated: 0 (not vaccinated), 1
(vaccinated)*/
},
    "meetingID": "test",
/*required, string, meeting number, range:[1,32]*/
    "PersonInfoExtends": [
/*optional, array, extended person information*/
{
    "id": 1,
/*optional, integer, extended person ID, range:[1,32]*/
        "value": "test"
/*optional, string, content of extended person information*/
},
    "name": "test",
/*optional, string, name*/
    "FaceRect": {
/*optional, object, rectangle for face picture*/
        "height": 1.000,
/*optional, float, height, range:[0.000,1.000]*/
        "width": 1.000,
/*optional, float, width, range:[0.000,1.000]*/
        "x": 0.000,
/*optional, float, horizontal coordinate in the upper-left corner, range:
[0.000,1.000]*/
        "y": 0.000
/*optional, float, vertical coordinate in the upper-left corner, range:
[0.000,1.000]*/
    }
},
    }
}
```

### F.4 JSON\_AcsEventTotalNum

AcsEventTotalNum message in JSON format

```
{  
    "AcsEventTotalNum":{  
        "totalNum":  
/*required, integer, total number of events that match the search conditions*/  
    }  
}
```

### F.5 JSON\_AcsEventTotalNumCond

AcsEventTotalNumCond message in JSON format

```
{  
    "AcsEventTotalNumCond":{  
        "major": ,  
/*required, integer, major type (the type value should be transformed to the  
decimal number), refer to Access Control Event Types for details*/  
        "minor": ,  
/*required, integer, minor type (the type value should be transformed to the  
decimal number), refer to Access Control Event Types for details*/  
        "startTime": "",  
/*optional, string, start time (UTC time), e.g., "2016-12-12T17:30:08+08:00"*/  
        "endTime": "",  
/*optional, string, end time (UTC time), e.g., "2017-12-12T17:30:08+08:00"*/  
        "cardNo": "",  
/*optional, string, card No.*/  
        "name": "",  
/*optional, string, cardholder name*/  
        "picEnable": ,  
/*optional, boolean, whether to contain pictures: "true"-yes, "false"-no*/  
        "beginSerialNo": ,  
/*optional, integer, start serial No.*/  
        "endSerialNo": ,  
/*optional, integer, end serial No.*/  
        "employeeNoString": "",  
/*optional, string, employee No. (person ID)*/  
        "eventAttribute": ""  
/*optional, string, event attribute: "attendance"-valid authentication,  
"other"*/  
    }  
}
```

#### See Also

[Access Control Event Types](#)

### F.6 JSON\_AddFaceRecordCond

Message about conditions of adding a face record, it is in JSON format

```
{  
    "faceURL": "",  
    /*optional, string type, picture storage URL inputted when uploading the face  
    picture by URL, the maximum length is 256 bytes*/  
    "faceLibType": "",  
    /*required, face picture library type: "blackFD"-list library, "staticFD"-  
    static library, string type, the maximum size is 32 bytes*/  
    "FDID": "",  
    /*required, face picture library ID, string type, the maximum size is 63  
    bytes*/  
    "FPID": "",  
    /*optional, string type, face record ID, which is the same as the employee No.  
(person ID), and the maximum length is 63 bytes*/  
    "name": "",  
    /*required, name of person in the face picture, string type, the maximum size  
    is 96 bytes*/  
    "gender": "",  
    /*optional, gender of person in the face picture: male, female, unknown, string  
    type, the maximum size is 32 bytes*/  
    "bornTime": "",  
    /*required, birthday of person in the face picture, ISO8601 time format, string  
    type, the maximum size is 20 bytes*/  
    "city": "",  
    /*optional, city code of birth for the person in the face picture, string type,  
    the maximum size is 32 bytes*/  
    "certificateType": "",  
    /*optional, string type, the max. size is 10 bytes, certificate type:  
    "officerID"-officer ID, "ID"-identify card, passport, other*/  
    "certificateNumber": "",  
    /*optional, certificate No., string, the max. size is 32 bytes*/  
    "caseInfo": "",  
    /*optional, case information, string type, the max. size is 192 bytes, it is  
    valid when faceLibType is "blackFD".*/  
    "tag": "",  
    /*optional, custom tag, up to 4 tags, which are separated by commas, string  
    type, the max. size is 195 bytes, it is valid when faceLibType is "blackFD".*/  
    "address": "",  
    /*optional, person address, string type, the max. size is 192 bytes, it is  
    valid when faceLibType is "staticFD".*/  
    "customInfo": "",  
    /*optional, custom information, string type, the max. size is 192 bytes, it is  
    valid when faceLibType is "staticFD".*/  
    "modelData": ""  
    /*optional, string type, target model data, non-modeled binary data needs to be  
    encrypted by Base64 during transmission*/  
    "transfer": true,  
    /*optional, boolean, whether to enable transfer*/  
    "operateType": "byTerminal",  
    /*optional, string, operation type: "byTerminal"-by terminal*/  
    "terminalNoList": [1],  
    /*optional, array, terminal ID list, this node is required when operation type  
    is "byTerminal"; currently, only one terminal is supported*/
```

```
"PicFeaturePoints": [
/*optional, array of object, feature points to be applied. If the device only
supports three types of feature points, when the platform applies more than
three types of feature points, the device will not return error information*/
    "featurePointType":"face",
/*required, string, feature point type: "face", "leftEye" (left eye),
"rightEye" (right eye), "leftMouthCorner" (left corner of mouth),
"rightMouthCorner" (right corner of mouth), "nose"*/
    "coordinatePoint":{
/*required, object, coordinates of the feature point*/
        "x":1,
/*required, int, normalized X-coordinate which is between 0 and 1000*/
        "y":1,
/*required, int, normalized Y-coordinate which is between 0 and 1000*/
        "width":1,
/*required, int, width which is between 0 and 1000. This node is required when
featurePointType is "face"*/
        "height":1
/*required, int, height which is between 0 and 1000. This node is required when
featurePointType is "face"*/
    }
},
"saveFacePic": true
/*optional, boolean, whether to save face pictures*/
}
```

## Remarks

If the field "faceURL" exists in the message, it indicates that the picture is uploaded via URL, and the "faceURL" of message should be set to picture URL. Otherwise, the picture is uploaded as binary data, which can be followed the message in JSON format, and separated by "boundary". See the example below.

## Example

### Add Face Record When Binary Picture is Uploaded in Form Format

```
1) POST /ISAPI/Intelligent/FDLib/FaceDataRecord?format=json
2) Accept: text/html, application/xhtml+xml,
3) Accept-Language: us-EN
4) Content-Type: multipart/form-data;
boundary=-----7e13971310878
5) User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;
Trident/5.0)
6) Accept-Encoding: gzip, deflate
7) Host: 10.10.36.29:8080
8) Content-Length: 9907
9) Connection: Keep-Alive
10) Cache-Control: no-cache
11)
12) -----7e13971310878
13) Content-Disposition: form-data; name="FaceDataRecord";
14) Content-Type: application/json
```

```
15) Content-Length: 9907
16)
17) {
a) "faceLibType": "blackFD",
b) "FDID": "1223344455566788",
c) "FPID": "11111aa",
d) "name": "Eric",
e) "gender": "male",
f) "bornTime": "2004-05-03",
g) "city": "130100",
h) "certificateType": "officerID",
i) "certificateNumber": "",
j) "caseInfo": "",
k) "tag": "aa,bb,cc,dd",
l) "address": "",
m) "customInfo": ""
18) }
19) -----7e13971310878
20) Content-Disposition: form-data; name="FaceImage";
21) Content-Type: image/jpeg
22) Content-Length: 9907
23)
24) .....JFIF.....`.....C..... .
25) ..
26) .....$.' ",#..(7),01444.'9=82<.342...C. ....
27) -----7e13971310878--
```



### Note

- In line 4, "Content-Type: multipart/form-data" indicates that the data is sent in form format. The "boundary" is a delimiter, you can assign value to it for distinguishing other ones.
  - In line 12, the request body consists of multiple same parts, and each part starts with "-" and from the customized "boundary" delimiter, the contents after the delimiter is the description of this part.
  - In line 13, "Content-Disposition" refers to condition parameters, when adding face record, the "name" must be set to "FaceDataRecord".
  - In line 14, "Content-Type" refers to JSON format, which based on UTF-8 character set.
  - In line 15, "Content-Length" refers to the size of data (contains the "\r\n" escape characters) from line 16 to line 18.
  - In line 16, the "\r\n\r\n" escape characters must be entered.
  - Line 19 is the start delimiter of next part.
  - Line 20 is the binary picture data, and the "name" must be set to "FaceImage".
  - Line 21 is the format of the binary picture data. Here, "image/jpeg" indicates that the following contents are JPEG format picture data.
  - In line 23, the "\r\n\r\n" escape characters must be entered.
  - In line 27, the customized "boundary" indicates the end of request body.
-

## F.7 JSON\_AddFaceRecordResult

Message about the result of adding the face record to face picture library, it is in JSON format.

```
{
  "requestURL": "",
  "statusCode": "",
  "statusString": "",
  "subStatusCode": "",
  "errorCode": "",
  "errorMsg": "",
  /*see the description of this node and above nodes in the message of
  JSON_ResponseStatus*/
  "FPID": ""
  /*optional, string type, face record ID returned when the face record is added,
  it is unique, and the maximum size is 63 bytes. This node is valid when
  errorCode is "1" and errorMsg is "ok"*/
}
```

### See Also

[JSON\\_ResponseStatus](#)

## F.8 JSON\_AntiSneakCfg

AntiSneakCfg message in JSON format

```
{
  "AntiSneakCfg": {
    "enable": ,
    /*required, boolean, whether to enable anti-passing back*/
    "startCardReaderNo": ,
    /*optional, integer, first card reader No., 0-no first card reader*/
  }
}
```

## F.9 JSON\_Attendance

JSON message about the parameters of attendance check by pressing the key

```
{
  "Attendance": {
    "enable": true,
    /*required, boolean, whether to enable*/
    "attendanceStatus": "",
    /*optional, string, attendance status: "checkIn"-check in, "checkOut"-check
    out, "breakOut"-break out, "breakIn"-break in, "overtimeIn"-overtime in,
    "overtimeOut"-overtime out*/
  }
}
```

```

    "label":"",
/*optional, string, custom name*/
}
}
```

## F.10 JSON\_AttendanceCap

JSON message about the configuration capability of attendance check by pressing the key

```
{
  "AttendanceCap": {
    "id": {
      /*required, int, key No. range*/
      "@min": 0,
      "@max": 0
    },
    "enable": {
      /*required, boolean, whether to enable*/
      "@opt": [true, false]
    },
    "label": {
      /*optional, string, custom name*/
      "@min": 0,
      "@max": 0
    },
    "attendanceStatus": {
      /*optional, string, attendance status*/
      "@opt": ["checkIn", "checkOut", "breakOut", "breakIn", "overtimeIn",
      "overtimeOut"]
    }
  }
}
```

## F.11 JSON\_AttendanceList

JSON message about the attendance parameter list

```
{
  "AttendanceList": [
    {
      "enable": true,
      /*required, boolean, whether to enable*/
      "attendanceStatus": "",
      /*optional, string, attendance status: "checkIn"-check in, "checkOut"-check
      out, "breakOut"-break out, "breakIn"-break in, "overtimeIn"-overtime in,
      "overtimeOut"-overtime out*/
      "label": ""
    }
  ]
}
```

```
    } ]  
}
```

### F.12 JSON\_AttendanceMode

JSON message about the attendance mode parameters

```
{  
    "AttendanceMode":{  
        "mode":"" ,  
        /*optional, string, attendance mode: "disable", "manual", "auto"-automatic,  
        "manualAndAuto"-manual and automatic*/  
        "attendanceStatusTime": 0 ,  
        /*optional, int, attendance status duration, unit: second. This node is valid  
        when mode is "manual" or "manualAndAuto"*/  
        "reqAttendanceStatus": true  
        /*optional, boolean, whether the attendance status is required*/  
    }  
}
```

### F.13 JSON\_AttendancePlanTemplate

JSON message about the parameters of the attendance schedule template

```
{  
    "AttendancePlanTemplate":{  
        "enable":true,  
        /*required, boolean, whether to enable: true-enable, false-disable*/  
        "property":"check",  
        /*required, string, attendance attribute: "check"-check in and check out,  
        "break"-break out and break in, "overtime"-overtime in and overtime out. Only  
        one attendance attribute can be configured for each template*/  
        "templateName":"" ,  
        /*required, string, template name*/  
        "weekPlanNo":1  
        /*required, int, week schedule No.*/  
    }  
}
```

### F.14 JSON\_AttendancePlanTemplateCap

JSON message about the configuration capability of the attendance schedule template

```
{  
    "AttendancePlanTemplateCap":{  
        "templateNo ":{  
        /*schedule template No.*/
```

```

        "@min":1,
        "@max":16
    },
    "property":{
/*required, attendance attribute: "check"-check in and check out, "break"-break
out and break in, "overtime"-overtime in and overtime out*/
        "@opt":["check", "break", "overtime"]
    },
    "enable":{
/*whether to enable: true-enable, false-disable*/
        "@opt":[true, false]
    },
    "templateName":{
/*template name*/
        "@min":1,
        "@max":32
    },
    "weekPlanNo":{
/*week schedule No.*/
        "@min":1,
        "@max":16
    },
    "holidayGroupNo":{

/*holiday group No.*/
        "@min":1,
        "@max":16
    }
}
}
}

```

### F.15 JSON\_AttendancePlanTemplateList

JSON message about the list of attendance schedule templates

```

{
    "AttendancePlanTemplateList":[{
        "templateNo":1,
/*required, int, schedule template No.*/
        "enable":true,
/*required, boolean, whether to enable: true-enable, false-disable*/
        "templateName":"",
/*required, string, template name*/
        "weekPlanNo":1
/*required, int, week schedule No.*/
    }]
}

```

## F.16 JSON\_AttendanceWeekPlan

JSON message about the parameters of the week attendance schedule

```
{
    "AttendanceWeekPlan": {
        "enable": true,
        /*required, boolean, whether to enable: true-enable, false-disable*/
        "WeekPlanCfg": [
            /*required, week schedule parameters*/
            {
                "id": 1,
                /*required, int, time period No.*/
                "week": "Monday",
                /*required, string, day of the week: "Monday", "Tuesday", "Wednesday",
                "Thursday", "Friday", "Saturday", "Sunday"*/
                "enable": true,
                /*required, boolean, whether to enable: true-enable, false-disable*/
                "TimeSegment": {
                    "beginTime": "10:10:00",
                    /*required, string, start time (device's local time)*/
                    "endTime": "12:10:00"
                    /*required, string, end time (device's local time)*/
                }
            }
        ]
    }
}
```

## F.17 JSON\_AttendanceWeekPlanCap

JSON message about the configuration capability of the week attendance schedule

```
{
    "AttendanceWeekPlanCap": {
        "planNo": {
            /*week attendance schedule No.*/
            "@min": 1,
            "@max": 16
        },
        "enable": {
            /*boolean, whether to enable: true-enable, false-disable*/
            "@opt": [true, false]
        },
        "WeekPlanCfg": {
            /*week schedule parameters*/
            "maxSize": 5,
            "id": {
                "@min": 1,
                "@max": 8
            },
            ...
        }
    }
}
```

```

    "week": {
        "@opt": ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"]
    },
    "enable": {
        /*boolean, whether to enable: true-enable, false-disable*/
        "@opt": [true, false]
    },
    "TimeSegment": {
        "beginTime": "",
        /*start time (device's local time)*/
        "endTime": "",
        /*end time (device's local time)*/
        "validUnit": "second"
        /*time accuracy: "hour", "minute", "second". If this node is not returned, the
time accuracy is "minute"*/
    }
}
}
}
}

```

### F.18 JSON\_BatchEditFaceRecord

Message about the condition of editing face records in the face picture library in a batch, and it is in JSON format.

```

{
    "faceURL": "",  

    /*optional, string type, picture storage URL inputted when uploading the face
picture by URL, the maximum length is 256 bytes*/
    "faceLibType": "",  

    /*required, string type, face picture library type: "blackFD"-list library,
"staticFD"-static library, the maximum length is 32 bytes*/
    "FDID": "",  

    /*required, string type, face picture library ID, the maximum length is 63
bytes, multiple face picture libraries should be separated by commas*/
    "FPID": "",  

    /*optional, string type, face record ID, it can be generated by the device or
inputted. If it is inputted, it should be the unique ID with the combination of
letters and digits, and the maximum length is 63 bytes; if it is generated by
the device automatically, it is the same as the employee No. (person ID)*/
    "name": "",  

    /*required, string type, name of the person in the face picture, the maximum
length is 96 bytes*/
    "gender": "",  

    /*optional, string type, gender of the person in the face picture: "male",
"female", "unknown", the maximum length is 32 bytes*/
    "bornTime": "",  

    /*required, string type, date of birth of the person in the face picture in
ISO8601 time format, the maximum length is 20 bytes*/
}

```

```
"city":"",
/*optional, string type, code of the city of birth for the person in the face
picture, the maximum length is 32 bytes*/
"certificateType":"",
/*optional, string type, ID type: "officerID"-officer ID, "ID"-ID card. The
maximum length is 10 bytes*/
"certificateNumber":"",
/*optional, string type, ID No., the maximum length is 32 bytes*/
"caseInfo":"",
/*optional, string type, case information, the maximum length is 192 bytes, it
is valid when faceLibType is "blackFD"*/
"tag":"",
/*optional, string type, custom tag, up to 4 tags can be added and they should
be separated by commas, the maximum length of each tag is 48 bytes, and the
maximum length of this node is 195 bytes. It is valid when faceLibType is
"blackFD"*/
"address":"",
/*optional, string type, person address, the maximum length is 192 bytes, it is
valid when faceLibType is "staticFD"*/
"customInfo":"",
/*optional, string type, custom information, the maximum length is 192 bytes,
it is valid when faceLibType is "staticFD"*/
"modelData":"",
/*optional, string type, target model data, non-modeled binary data needs to be
encrypted by base64 during transmission*/
"operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal*/
"terminalNoList": [1],
/*optional, array, terminal ID list, this node is required when operation type
is "byTerminal"; currently, only one terminal is supported*/
"PicFeaturePoints": [
/*optional, array of object, feature points to be applied. If the device only
supports three types of feature points, when the platform applies more than
three types of feature points, the device will not return error information*/
    "featurePointType": "face",
    /*required, string, feature point type: "face", "leftEye" (left eye),
    "rightEye" (right eye), "leftMouthCorner" (left corner of mouth),
    "rightMouthCorner" (right corner of mouth), "nose"*/
        "coordinatePoint": {
            /*required, object, coordinates of the feature point*/
                "x": 1,
                /*required, int, normalized X-coordinate which is between 0 and 1000*/
                "y": 1,
                /*required, int, normalized Y-coordinate which is between 0 and 1000*/
                "width": 1,
                /*required, int, width which is between 0 and 1000. This node is required when
featurePointType is "face"*/
                "height": 1
            /*required, int, height which is between 0 and 1000. This node is required when
featurePointType is "face"*/
        }
    ],
}
```

```

    "saveFacePic": true
/*optional, boolean, whether to save face pictures*/
}

```

### **F.19 JSON\_BlackBodyCfg**

JSON message about the black body parameters

```

{
  "enabled":true,
/*required, boolean, whether to enable the black body*/
  "Position":{
/*optional, object, black body position (coordinate), the value is normalized
to a number between 0 and 1000*/
    "x":1,
/*optional, int, X-coordinate, value range: [0,1000]*/
    "y":1
/*optional, int, Y-coordinate, value range: [0,1000]*/
  },
  "distance":1.0,
/*optional, float, distance between the black body and the lens, the value is
accurate to one decimal place, value range: [0.0,10.0], unit: meter*/
  "emissivity":0.10,
/*optional, float, emissivity, the value is accurate to two decimal places,
value range: [0.00,1.00]*/
  "unit":"celsius",
/*optional, string, temperature unit: "celsius", "fahrenheit"*/
  "temperature":30.0
/*optional, float, black body temperature. When the value of the node unit is
"celsius", the value of this node is between 30.0 and 50.0; when the value of
the node unit is "fahrenheit", the value of this node is between 86.0 and
122.0. The value is accurate to one decimal place*/
}

```

### **F.20 JSON\_Cap\_AcsCfg**

AcsCfg capability message in JSON format

```

{
  "AcsCfg":{
    "RS485Backup":"true,false",
/*optional, boolean, whether to enable downstream RS-485 communication
redundancy: "true"-yes, "false"-no*/
    "showCapPic":"true,false",
/*optional, boolean, whether to display the captured picture: "true"-yes,
"false"-no*/
    "showUserInfo":"true,false",
/*optional, boolean, whether to display user information: "true"-yes, "false"-no*/
}

```

```

    "overlayUserInfo":"true,false",
/*optional, boolean, whether to overlay user information: "true"-yes, "false"-no*/
    "voicePrompt":"true,false",
/*optional, boolean, whether to enable audio announcement: "true"-yes, "false"-no*/
    "uploadCapPic":"true,false",
/*optional, boolean, whether to upload the picture from linked capture: "true"-yes, "false"-no*/
    "saveCapPic":"true,false",
/*optional, boolean, whether to save the capture picture: "true"-yes, "false"-no*/
    "inputCardNo":"true,false",
/*optional, boolean, whether to allow inputting card No. on keypad: "true"-yes, "false"-no*/
    "enableWifiDetect":"true,false",
/*optional, boolean, whether to enable Wi-Fi probe: "true"-yes, "false"-no*/
    "enable3G4G":"true,false",
/*optional, boolean, whether to enable 3G/4G: "true"-yes, "false"-no*/
    "protocol":{
/*optional, string, communication protocol type of the card reader: "Private"-private protocol, "OSDP"-OSDP protocol*/
        "@opt":"Private,OSDP"
    },
    "enableCaptureCertificate": "true,false",
/*optional, string, whether to enable capturing the ID picture: true (yes), false (no). The captured ID picture will be compared with the captured face picture to check whether it is the same person. If this node does not exist, it indicates that this function is not supported*/
    "showPicture":"true,false",
/*optional, boolean, whether to display the authenticated picture: "true"-display, "false"-not display*/
    "showEmployeeNo":"true,false",
/*optional, boolean, whether to display the authenticated employee ID: "true"-display, "false"-not display*/
    "showName":"true,false",
/*optional, boolean, whether to display the authenticated name: "true"-display, "false"-not display*/
    "desensitiseEmployeeNo":{
/*dependent, boolean, whether to enable employee No. de-identification for local UI display: true (yes), false (no). This node is valid when the value of the node showEmployeeNo is true*/
        "@opt": [true,false]
    },
    "desensitiseName":{
/*dependent, boolean, whether to enable name de-identification for local UI display: true (yes), false (no). This node is valid when the value of the node showName is true*/
        "@opt": [true,false]
    },
    "thermalEnabled": {
/*optional, boolean, whether to enable temperature measurement: true-enable

```

```

(default), false-disable*/
    "@opt": [true,false]
},
"thermalMode": {
/*optional, boolean, whether to enable temperature measurement only mode: true-
enable (only for temperature measurement), false-disable (default)*/
    "@opt": [true,false]
},
"thermalPictureEnabled": {
/*optional, boolean, whether to enable uploading visible light pictures in
temperature measurement only mode: true-enable, false-disable (default). This
field is used to control uploading captured pictures and visible light
pictures*/
    "@opt": [true,false]
},
"isSupportThermalIp": true,
/*optional, boolean, whether it supports configuring IP address of the
thermography device: true-yes, this field is not returned-no*/
    "highestThermalThreshold": {
/*optional, float, upper limit of the temperature threshold which is accurate
to one decimal place, unit: Celsius*/
        "@min": ,
        "@max":
    },
    "lowestThermalThreshold": {
/*optional, float, lower limit of the temperature threshold which is accurate
to one decimal place, unit: Celsius*/
        "@min": ,
        "@max":
    },
    "thermalDoorEnabled": {
/*optional, boolean, whether to open the door when the temperature is above the
upper limit (highestThermalThreshold) or below the lower limit
(lowestThermalThreshold) of the threshold: true-open the door, false-not open
the door (default)*/
        "@opt": [true,false]
},
    "QRCodeEnabled": {
/*optional, boolean, whether to enable QR code function: true-enable, false-
disable (default)*/
        "@opt": [true,false]
},
    "remoteCheckDoorEnabled": {
/*optional, boolean, whether to enable controlling the door by remote
verification: true-control, false-not control (default)*/
        "@opt": [true,false]
},
    "checkChannelType": {
/*dependent, string, verification channel type: "Ezviz"-EZVIZ channel, "ISUP"-_
ISUP channel, "ISAPI"-ISAPI channel, "PrivateSDK"-private SDK channel,
"ISAPIListen"-ISAPI listening channel. This field is valid when
remoteCheckDoorEnabled is true*/
}

```

```

    "@opt": ["Ezviz", "ISUP", "ISAPI", "PrivateSDK", "ISAPIListen"]
},
"isSupportChannelIp": true,
/*optional, boolean, whether it supports configuring IP address of the verification channel: true-yes, this field is not returned-no*/
"uploadVerificationPic":"",
/*optional, boolean, whether to upload the authenticated picture: true, false*/
"saveVerificationPic":"",
/*optional, boolean, whether to save the authenticated picture: true, false*/
"saveFacePic":"",
/*optional, boolean, whether to save the registered face picture: true, false*/
"thermalUnit":{
/*optional, object, temperature unit: "celsius" (default), "fahrenheit"*/
"@opt":["celsius", "fahrenheit"]
},
"highestThermalThresholdF":{
/*optional, object, the maximum value of the temperature threshold. The value is accurate to one decimal place, and the unit is Fahrenheit*/
"@min":1.0,
"@max":1.0
},
"lowestThermalThresholdF":{
/*optional, object, the minimum value of the temperature threshold. The value is accurate to one decimal place, and the unit is Fahrenheit*/
"@min":1.0,
"@max":1.0
},
"thermalCompensation":{
/*optional, object, temperature compensation, the value is accurate to one decimal place. The unit depends on the node thermalUnit. If the node thermalUnit does not exist, the default unit is Celsius*/
"@min":-99.9,
"@max":99.9
}
}
}
}

```

## F.21 JSON\_Cap\_AcsEvent

AcsEvent capability message in JSON format

```
{
  "AcsEvent": {
    "AcsEventCond": {
      /*optional, search conditions*/
      "searchID": {
        /*required, string type, search ID, which is used to confirm the upper-level platform or system. If the platform or the system is the same one during two searching, the search history will be saved in the memory to speed up next searching*/
      }
    }
  }
}
```

```

        "@min": ,
        "@max": ,
    },
    "searchResultPosition":{
/*required, integer, the start position of the search result in the result
list. When there are multiple records and you cannot get all search results at
a time, you can search for the records after the specified position next time*/
        "@min": ,
        "@max": ,
    },
    "maxResults":{
/*required, integer, maximum number of search results*/
        "@min": ,
        "@max": ,
    },
    "major":{
/*required, integer, major alarm/event types (the type value should be
transformed to the decimal number), refer to Access Control Event Types for
details*/
        "@opt": "0,1,2,3,5"
    },
    "minorAlarm":{
/*required, integer, minor alarm type (the type value should be transformed to
the decimal number), refer to Access Control Event Types for details*/
        "@opt": "1024,1025,1026,1027..."
    },
    "minorException":{
/*required, integer, minor exception type (the type value should be transformed
to the decimal number), refer to Access Control Event Types for details*/
        "@opt": "39,58,59,1024..."
    },
    "minorOperation":{
/*required, integer, minor operation type (the type value should be transformed
to the decimal number), refer to Access Control Event Types for details*/
        "@opt": "80,90,112,113..."
    },
    "minorEvent":{
/*required, integer, minor event type (the type value should be transformed to
the decimal number), refer to Access Control Event Types for details*/
        "@opt": "1,2,3,4..."
    },
    "startTime":{
/*optional, string, start time (UTC time)*/
        "@min": ,
        "@max": ,
    },
    "endTime":{
/*optional, string, end time (UTC time)*/
        "@min": ,
        "@max": ,
    },
    "cardNo":{

```

```
/*optional, string, card No.*/
    "@min": ,
    "@max":
},
"name":{
/*optional, string, cardholder name*/
    "@min": ,
    "@max":
},
"picEnable": "true,false",
/*optional, boolean, whether to contain pictures: "false"-no, "true"-yes*/
"beginSerialNo":{
/*optional, integer, start serial No.*/
    "@min": ,
    "@max":
},
"endSerialNo":{
/*optional, integer, end serial No.*/
    "@min": ,
    "@max":
},
"employeeNoString":{
/*optional, string, employee No. (person ID)*/
    "@min": ,
    "@max":
},
"eventAttribute":{
/*optional, string, event attribute: "attendance"-valid authentication,
"other"*/
    "@opt":"attendance,other"
},
"employeeNo": {
/*optional, string, employee No. (person ID)*/
    "@min": ,
    "@max":
},
"timeReverseOrder": "true,false",
/*optional, boolean, whether to return events in descending order of time
(later events will be returned first): true-yes, false or this node is not
returned-no*/
"isAbnormalTemperature":{
/*optional, object, whether the skin-surface temperature is abnormal*/
    "@opt":[true, false]
/*optional, array of boolean, options: true (yes), false (no)*/
}
},
"InfoList":{
/*optional, event details*/
    "maxSize": 10,
    "time":{
/*required, string, time (UTC time)*/
    "@min": ,
```

```

        "@max":  
    },  
    "netUser":{  
/*optional, string, user name*/  
        "@min": ,  
        "@max":  
    },  
    "remoteHostAddr":{  
/*optional, string, remote host address*/  
        "@min": ,  
        "@max":  
    },  
    "cardNo":{  
/*optional, string, card No.*/  
        "@min": ,  
        "@max":  
    },  
    "cardType":{  
/*optional, integer, card type: "1"-normal card, "2"-disabled card, "3"-  
blocklist card, "4"-patrol card, "5"-duress card, "6"-super card, "7"-visitor  
card, "8"-dismiss card*/  
        "@opt": "1,2,3,4,5,6,7,8"  
    },  
    "whiteListNo":{  
/*optional, integer, allowlist No., which is between 1 and 8*/  
        "@min": ,  
        "@max":  
    },  
    "reportChannel":{  
/*optional, integer, channel type for uploading alarm/event: "1"-for uploading  
arming information, "2"-for uploading by central group 1, "3"-for uploading by  
central group 2*/  
        "@opt": "1,2,3"  
    },  
    "cardReaderKind":{  
/*optional, integer, authentication unit type: "1"-IC card reader, "2"-ID card  
reader, "3"-QR code scanner, "4"-fingerprint module*/  
        "@opt": "1,2,3,4"  
    },  
    "cardReaderNo":{  
/*Optional, integer, authentication unit No.*/  
        "@min": ,  
        "@max":  
    },  
    "doorNo":{  
/*optional, integer, door or floor No.*/  
        "@min": ,  
        "@max":  
    },  
    "verifyNo":{  
/*optional, integer, multiple authentication No.*/  
        "@min": ,

```

```

        "@max":  
    },  
    "alarmInNo":{  
/*optional, integer, alarm input No.*/  
        "@min": ,  
        "@max":  
    },  
    "alarmOutNo":{  
/*optional, integer, alarm output No.*/  
        "@min": ,  
        "@max":  
    },  
    "caseSensorNo":{  
/*optional, integer, event trigger No.*/  
        "@min": ,  
        "@max":  
    },  
    "RS485No":{  
/*optional, integer, RS-485 channel No.*/  
        "@min": ,  
        "@max":  
    },  
    "multiCardGroupNo":{  
/*optional, integer, group No.*/  
        "@min": ,  
        "@max":  
    },  
    "accessChannel":{  
/*optional, integer, swing barrier No.*/  
        "@min": ,  
        "@max":  
    },  
    "deviceNo":{  
/*optional, integer, device No.*/  
        "@min": ,  
        "@max":  
    },  
    "distractControlNo":{  
/*optional, integer, distributed access controller No.*/  
        "@min": ,  
        "@max":  
    },  
    "employeeNo":{  
/*optional, string, employee No. (person ID)*/  
        "@min": ,  
        "@max":  
    },  
    "localControllerID":{  
/*optional, integer, distributed access controller No.: "0"-access controller,  
"1" to "64"-distributed access controller No.1 to distributed access controller  
No.64*/  
        "@min": ,

```

```

        "@max": ,
    },
    "InternetAccess":{
/*optional, integer, network interface No.: "1"-upstream network interface No.
1, "2"-upstream network interface No.2, "3"-downstream network interface No.1*/
        "@min": ,
        "@max": ,
    },
    "type":{
/*optional, integer, zone type: "0"-instant alarm zone, "1"-24-hour alarm zone,
"2"-delayed zone, "3"-internal zone, "4"-key zone, "5"-fire alarm zone, "6"-perimeter protection, "7"-24-hour slient alarm zone, "8"-24-hour auxiliary zone, "9"-24-hour shock alarm zone, "10"-emergency door open alarm zone, "11"-emergency door closed alarm zone, "255"-none*/
        "@opt": "0,1,2,3,4,5,6,7,8,9,10,11,255"
    },
    "MACAddr":{
/*optional, string, physical address*/
        "@min": ,
        "@max": ,
    },
    "swipeCardType":{
/*optional, integer, card swiping type: "0"-invalid, "1"-QR code*/
        "@opt": "0,1"
    },
    "serialNo":{
/*optional, integer, event serial No., which is used to judge whether the event loss occurred*/
        "@min": ,
        "@max": ,
    },
    "channelControllerID":{
/*optional, integer, lane controller No.: "1"-main lane controller, "2"-sub lane controller*/
        "@opt": "0,1"
    },
    "channelControllerLampID":{
/*optional, integer, light board No. of lane controller, which is between 1 and 255*/
        "@min": ,
        "@max": ,
    },
    "channelControllerIRAdaptorID":{
/*optional, integer, IR adapter No. of lane controller, which is between 1 and 255*/
        "@min": ,
        "@max": ,
    },
    "channelControllerIREmitterID":{
/*optional, integer, active infrared intrusion detector No. of lane controller, which is between 1 and 255*/
        "@min": ,
    }
}

```

```

        "@max":  
    },  
    "userType":{  
/*optional, string, person types: "normal"-normal person (household), "visitor"-  
visitor, "blacklist"-person in blocklist, "administrators"-administrator*/  
        "@opt": "normal,visitor,blackList,administrators"  
    },  
    "currentVerifyMode": {  
/*optional, string, authentication modes: "cardAndPw"-card+password, "card",  
"cardOrPw"-card or password, "fp"-fingerprint, "fpAndPw"-fingerprint+password,  
"fpOrCard"-fingerprint or card, "fpAndCard"-fingerprint+card, "fpAndCardAndPw"-  
fingerprint+card+password, "faceOrFpOrCardOrPw"-face or fingerprint or card or  
password, "faceAndFp"-face+fingerprint, "faceAndPw"-face+password,  
"faceAndCard"-face+card, "face", "employeeNoAndPw"-employee No.+password,  
"fpOrPw"-fingerprint or password, "employeeNoAndFp"-employee No.+fingerprint,  
"employeeNoAndFpAndPw"-employee No.+fingerprint+password, "faceAndFpAndCard"-  
face+fingerprint+card, "faceAndPwAndFp"-face+password+fingerprint,  
"employeeNoAndFace"-employee No.+face, "faceOrfaceAndCard"-face or face+card,  
"fpOrface"-fingerprint or face, "cardOrfaceOrPw"-card or face or password,  
"cardOrFpOrPw"-card or fingerprint or password*/  
        "@opt":  
"cardAndPw,card,cardOrPw,fp,fpAndPw,fpOrCard,fpAndCard,fpAndCardAndPw,faceOrFpOr  
CardOrPw,faceAndFp,faceAndPw,faceAndCard,face,employeeNoAndPw,fpOrPw,employeeNoA  
ndFp,employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndFace,face  
OrfaceAndCard,fpOrface,cardOrfaceOrPw,cardOrFpOrPw"  
    },  
    "QRCodeInfo":{  
/*optional, object, QR code information*/  
        "@min":1,  
        "@max":1  
    },  
    "thermometryUnit": {  
/*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheitz"-  
Fahrenheit, "kelvin"-Kelvin*/  
        "@opt": ["celsius","fahrenheitz","kelvin"]  
    },  
    "currTemperature": {  
/*optional, float, face temperature which is accurate to one decimal place*/  
        "@min":1 ,  
        "@max":1  
    },  
    "isAbnormalTemperature": {  
/*optional, boolean, whether the face temperature is abnormal: true-yes, false-  
no*/  
        "@opt": [true,false]  
    },  
    "RegionCoordinates": {  
/*optional, face temperature's coordinates*/  
        "positionX": {  
/*optional, int, normalized X-coordinate which is between 0 and 1000*/  
            "@min": 0,  
            "@max": 1000

```

```

        },
        "positionY": {
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
            "@min": 0,
            "@max": 1000
        }
    },
    "picEnable": "true,false",
/*optional, boolean, whether to contain pictures*/
    "picturesNumber":{
/*optional, integer, number of captured pictures if the capture linkage action
is configured. This node will be 0 or not be returned if there is no picture*/
        "@min": ,
        "@max":
    },
    "filename": {
/*optional, string, file name. If multiple pictures are returned at a time, the
file name of each picture should be unique, and the value of this node should
be the same as the following one*/
        "@min": ,
        "@max":
    },
    "attendanceStatus":{
/*optional, string, attendance status: "undefined", "checkIn"-check in,
"checkOut"-check out, "breakOut"-break out, "breakIn"-break in, "overtimeIn"-overtime
in, "overTimeOut"-overtime out*/
        "@opt":"undefined,checkIn,checkOut,breakOut,breakIn,overtimeIn,overtimeOut"
    },
    "label":{
/*optional, string, custom attendance name*/
        "@min": ,
        "@max":
    },
    "statusValue":{
/*optional, integer, status value*/
        "@min":0,
        "@max":255
    },
    "mask": {
/*optional, string, whether the person is wearing mask: "unknown", "yes"-wearing
mask, "no"-not wearing mask*/
        "@opt": "unknown,yes,no"
    },
    "pictureURL":{
/*optional, object, URL of the captured picture*/
        "@min":1,
        "@max":1
    },
    "helmet": {
/*optional, string, whether the person is wearing hard hat: "unknown", "yes"-wearing
hard hat, "no"-not wearing hard hat*/

```

```
        "@opt": "unknown, yes, no"
    },
    "visibleLightPicUrl": {
/*optional, object, URL of the visible light picture*/
        "@min": 1,
        "@max": 1
    },
    "thermalPicUrl": {
/*optional, object, URL of the thermal picture*/
        "@min": 1,
        "@max": 1
    },
    "HealthInfo": {
/*optional, object, health information*/
        "healthCode": {
/*optional, object, health code status*/
            "@opt": [0, 1, 2, 3, 4, 5, 6]
/*optional, array of int, options: 0 (no request), 1 (no health code), 2 (green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6 (other error, e.g., searching failed due to API exception), 7 (searching for the health code timed out)*/
        },
        "NADCode": {
/*optional, object, nucleic acid test result: 0 (no result), 1 (negative, which means normal), 2 (positive, which means diagnosed), 3 (the result has expired)*/
            "@opt": [0, 1, 2, 3]
        },
        "travelCode": {
/*optional, object, trip code: 0 (no trip in the past 14 days), 1 (once left in the past 14 days), 2 (has been to the high-risk area in the past 14 days), 3 (other)*/
            "@opt": [0, 1, 2, 3]
        },
        "vaccineStatus": {
/*optional, object, whether the person is vaccinated: 0 (not vaccinated), 1 (vaccinated)*/
            "@opt": [0, 1]
        }
    }
}
```

### See Also

[Access Control Event Types](#)

## F.22 JSON\_Cap\_AcsEventTotalNum

AcsEventTotalNum capability message in JSON format

```
{
    "AcsEvent": {
        "AcsEventTotalNumCond": {
            /*optional, search conditions*/
            "major": {
                /*required, integer type, major type (the type value should be transformed to
                the decimal number): 0-all, 1-major alarm type, 2-major exception type, 3-major
                operation type, 5-major event type, refer to
                Access Control Event Types
                for details*/
                "@opt": "0,1,2,3,5"
            },
            "minorAlarm": {
                /*required, integer, minor alarm type (the type value should be transformed to
                the decimal number), refer to Access Control Event Types for details*/
                "@opt": "1024,1025,1026,1027..."
            },
            "minorException": {
                /*required, integer, minor exception type (the type value should be transformed
                to the decimal number), refer to Access Control Event Types for details*/
                "@opt": "39,58,59,1024..."
            },
            "minorOperation": {
                /*required, integer, minor operation type (the type value should be transformed
                to the decimal number), refer to Access Control Event Types for details*/
                "@opt": "80,90,112,113..."
            },
            "minorEvent": {
                /*required, integer, minor event type (the type value should be transformed to
                the decimal number), refer to Access Control Event Types for details*/
                "@opt": "1,2,3,4..."
            },
            "startTime": {
                /*optional, string, start time (UTC time)*/
                "@min": ,
                "@max":
            },
            "endTime": {
                /*optional, string, end time (UTC time)*/
                "@min": ,
                "@max":
            },
            "cardNo": {
                /*optional, string, card No.*/
                "@min": ,
                "@max":
            },
            "name": {
                /*optional, string, cardholder name*/
                "@min": ,
                "@max":
            }
        }
    }
}
```

```

        "picEnable": "true, false",
/*optional, boolean, whether to contain pictures: "false"-no, "true"-yes*/
        "beginSerialNo": {
/*optional, integer, start serial No.*/
            "@min": ,
            "@max":
        },
        "endSerialNo": {
/*optional, integer, end serial No.*/
            "@min": ,
            "@max":
        },
        "employeeNoString": {
/*optional, string, employee No. (person ID)*/
            "@min": ,
            "@max":
        },
        "eventAttribute": {
/*optional, string, event attribute: "attendance"-valid authentication,
"other"*/
            "@opt": "attendance, other"
        }
    },
    "totalNum": {
/*required, integer, total number of events that match the search conditions*/
        "@min": ,
        "@max":
    }
}
}

```

### See Also

[Access Control Event Types](#)

## F.23 JSON\_Cap\_AntiSneakCfg

AntiSneakCfg capability message in JSON format

```

{
    "AntiSneakCfg": {
        "enable": "true, false",
/*required, boolean, whether to enable anti-passing back*/
        "startCardReaderNo": {
/*optional, integer, first card reader No., 0-no first card reader*/
            "@min": 1,
            "@max": 4
        }
    }
}

```

## F.24 JSON\_Cap\_AttendanceMode

JSON message about the configuration capability of the attendance mode

```
{  
    "AttendanceMode": {  
        "mode": {  
            /*optional, string, attendance mode: "disable", "manual", "auto"-automatic,  
            "manualAndAuto"-manual and automatic*/  
            "@opt": ["disable", "manual", "auto", "manualAndAuto"]  
        },  
        "attendanceStatusTime": {  
            /*optional, int, attendance status duration, unit: second. This node is valid  
            when mode is "manual" or "manualAndAuto"*/  
            "@min": 0,  
            "@max": 0  
        }  
    }  
}
```

## F.25 JSON\_Cap\_BlackBodyCfg

JSON message about the configuration capability of the black body

```
{  
    "enabled": {  
        /*required, object, whether to enable the black body. The black body is used to  
        calibrate the temperature of the thermography module. You need to put the black  
        body with fixed temperature in front of the device and calibrate the  
        temperature of the thermography module according to the black body in the  
        image*/  
        "@opt": [true, false]  
    },  
    "Position": {  
        /*optional, object, black body position (coordinate), the value is normalized  
        to a number between 0 and 1000*/  
        "x": {  
            /*optional, object, X-coordinate*/  
            "@min": 0,  
            "@max": 1000  
        },  
        "y": {  
            /*optional, object, Y-coordinate*/  
            "@min": 0,  
            "@max": 1000  
        }  
    },  
    "distance": {  
        /*optional, object, distance between the black body and the lens, the value is  
        */  
    }  
}
```

```

accurate to one decimal place, unit: meter*/
    "@min":0.0,
    "@max":10.0
},
"emissivity":{
/*optional, object, emissivity, the value is accurate to two decimal places.
The emissivity is applied from the system or platform to the device and is
transmitted to the thermography module by the device for temperature
measurement*/
    "@min":0.00,
    "@max":1.00
},
"TemperatureList":[{
/*optional, array of object, temperature list of the black body. The second
decimal place of the black body's temperature will be rounded to eliminate the
error. For example, 0.95 will be input as 1.0*/
    "unit":"celsius",
/*optional, string, temperature unit: "celsius", "fahrenheit"*/
    "temperature":{
/*optional, object, black body temperature. When the unit is "celsius", the
value of this node is between 30.0 and 50.0; when the unit is "fahrenheit", the
value of this node is between 86.0 and 122.0*/
        "@min":30.0,
        "@max":50.0
    }
}
}]
}

```

## F.26 JSON\_Cap\_CardInfo

CardInfo capability message in JSON format

```
{
  "CardInfo":{
    "supportFunction":{
      /*required, supported functions of adding, editing, deleting, searching for
card information, and getting the total number of added cards: "post"-add,
"delete", "put"-edit, "get"-search, "setUp"-set*/
        "@opt":"post,delete,put,get,setUp"
    },
    "CardInfoSearchCond":{
      /*optional, search conditions*/
        "searchID":{
          /*required, string type, search ID, which is used to check the upper-level
platform or system. If the platform or the system is the same one during two
searching, the search history will be saved in the memory to speed up next
searching*/
            "@min":1,
            "@max":36
        },
    }
}
```

```
        "maxResults":{  
/*required, integer32, maximum number of obtained records*/  
        "@min":1,  
        "@max":30  
    },  
    "EmployeeNoList":{  
/*optional, person ID list*/  
        "maxSize":56,  
        "employeeNo":{  
/*optional, string, employee No. (person ID)*/  
            "@min": ,  
            "@max":  
        }  
    },  
    "CardNoList":{  
/*optional, card No. list*/  
        "maxSize":56,  
        "cardNo":{  
/*optional, string, card No.*/  
            "@min":1,  
            "@max":32  
        }  
    },  
    "CardInfoDelCond":{  
/*optional, deleting conditions*/  
        "EmployeeNoList":{  
/*optional, person ID list*/  
            "maxSize":56,  
            "employeeNo":{  
/*optional, string, employee No. (person ID)*/  
                "@min": ,  
                "@max":  
            }  
        },  
        "CardNoList":{  
/*optional, card No. list*/  
            "maxSize":56,  
            "cardNo":{  
/*optional, string, card No.*/  
                "@min":1,  
                "@max":32  
            }  
        },  
        "cardNo":{  
/*required, string, card No.*/  
            "@min":1,  
            "@max":32  
        },  
        "employeeNo":{  
/*required, string, employee No. (person ID)*/  
    }
```

```

        "@min": ,
        "@max":
    },
    "cardType":{
/*required, string, card type: "normalCard"-normal card, "patrolCard"-patrol
card, "hijackCard"-duress card, "superCard"-super card, "dismissingCard"-dismiss
card, "emergencyCard"-emergency card (it is used to assign permission to a temporary
card, but it cannot open the door)*/

"@opt":"normalCard,patrolCard,hijackCard,superCard,dismissingCard,emergencyCard"
},
    "leaderCard":{
/*optional, string, whether to support first card authentication function*/
        "@min":1,
        "@max":32
},
    "checkCardNo":"true,false",
/*optional, boolean, whether to enable duplicated card verification: "false"-disable,
"true"-enable. If this node is not configured, the device will verify the duplicated
card by default. When there is no card information, you can set checkCardNo to "false"
to speed up data applying; otherwise, it is not recommended to configure this node*/
    "checkEmployeeNo":"true,false",
/*optional, boolean, whether to check the existence of the employee No. (person
ID): "false"-no, "true"-yes. If this node is not configured, the device will judge the
existence of the employee No. (person ID) by default. If this node is set to "false",
the device will not judge the existence of the employee No. (person ID) to speed up
data applying; if this node is set to "true" or NULL, the device will judge the
existence of the employee No. (person ID), and it is recommended to set this node to
"true" or NULL if there is no need to speed up data applying*/
    "addCard":"true,false",
/*optional, boolean type, whether to add the card if the card information being
edited does not exist: "false"-no (if the device has checked that the card
information being edited does not exist, the failure response message will be
returned along with the error code), "true"-yes (if the device has checked that
the card information being edited does not exist, the success response message
will be returned, and the card will be added). If this node is not configured,
the card will not be added by default*/
    "maxRecordNum":
/*required, integer type, supported maximum number of records (card records)*/
}
}

```

## F.27 JSON\_Cap\_CardReaderAntiSneakCfg

CardReaderAntiSneakCfg capability message in JSON format

```
{
    "CardReaderAntiSneakCfg": {

```

```

"cardReaderNo": {
/*optional, string, card reader No.*/
    "@min": ,
    "@max": 
}
"enable": "true,false",
/*required, boolean, whether to enable the anti-passing back function of the
card reader: "true"-enable, "false"-disable*/
"followUpCardReader": {
/*optional, array, following card reader No. after the first card reader*/
    "@min": ,
    "@max": 
}
}
}

```

## F.28 JSON\_Cap\_CardReaderCfg

CardReaderCfg capability message in JSON format

```

{
    "CardReaderCfg": {
        "cardReaderNo": {
/*optional, integer, card reader No.*/
            "@min": ,
            "@max": 
        },
        "enable": "true,false",
/*required, boolean, whether to enable: "true"-yes, "false"-no*/
        "okLedPolarity": {
/*optional, string, OK LED polarity: "cathode", "anode"*/
            "@opt": "cathode,anode"
        },
        "errorLedPolarity": {
/*optional, string, error LED polarity: "cathode", "anode"*/
            "@opt": "cathode,anode"
        },
        "buzzerPolarity": {
/*optional, string, buzzer polarity: "cathode", "anode"*/
            "@opt": "cathode,anode"
        },
        "swipeInterval": {
/*optional, integer, time interval of repeated authentication, which is valid
for authentication modes such as fingerprint, card, face, etc., unit: second*/
            "@min": 1,
            "@max": 255
        },
        "pressTimeout": {
/*optional, integer, timeout to reset entry on keypad, unit: second*/
            "@min": 1,

```

```

        "@max":255
    },
    "enableFailAlarm":"true,false",
/*optional, boolean, whether to enable excessive failed authentication attempts
alarm*/
    "maxReadCardFailNum":{
/*optional, integer, maximum number of failed authentication attempts*/
        "@min":1,
        "@max":255
    },
    "enableTamperCheck":"true,false",
/*optional, boolean, whether to enable tampering detection*/
    "offlineCheckTime":{
/*optional, integer, time to detect after the card reader is offline, unit:
second*/
        "@min":1,
        "@max":255
    },
    "fingerPrintCheckLevel":{
/*optional, integer, fingerprint recognition level: 1-1/10 false acceptance
rate (FAR), 2-1/100 false acceptance rate (FAR), 3-1/1000 false acceptance rate
(FAR), 4-1/10000 false acceptance rate (FAR), 5-1/100000 false acceptance rate
(FAR), 6-1/1000000 false acceptance rate (FAR), 7-1/10000000 false acceptance
rate (FAR), 8-1/100000000 false acceptance rate (FAR), 9-3/100 false acceptance
rate (FAR), 10-3/1000 false acceptance rate (FAR), 11-3/10000 false acceptance
rate (FAR), 12-3/100000 false acceptance rate (FAR), 13-3/1000000 false
acceptance rate (FAR), 14-3/10000000 false acceptance rate (FAR),
15-3/100000000 false acceptance rate (FAR), 16-Automatic Normal, 17-Automatic
Secure, 18-Automatic More Secure (currently not support)*/
        "@opt":"1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18"
    },
    "useLocalController":"true,false",
/*ro, opt, boolean, whether it is connected to the distributed controller*/
    "localControllerID":{
/*ro, opt, integer, distributed controller No., which is between 1 and 64, 0-
unregistered. This field is valid only when useLocalController is "true"*/
        "@min":0,
        "@max":64
    },
    "localControllerReaderID":{
/*ro, opt, integer, card reader ID of the distributed controller, 0-
unregistered. This field is valid only when useLocalController is "true"*/
        "@min":0,
        "@max":4
    },
    "cardReaderChannel":{
/*ro, opt, integer, communication channel No. of the card reader: 0-Wiegand or
offline, 1-RS-485A, 2-RS-485B. This field is valid only when useLocalController
is "true"*/
        "@opt":"0,1,2"
    },
    "fingerPrintImageQuality":{


```

```

/*opt, integer, fingerprint image quality: 1-low quality (V1), 2-medium quality
(V1), 3-high quality (V1), 4-highest quality (V1), 5-low quality (V2), 6-medium
quality (V2), 7-high quality (V2), 8-highest quality (V2)*/
    "@opt":"1,2,3,4,5,6,7,8"
},
"fingerPrintContrastTimeOut":{
/*optional, integer, fingerprint comparison timeout, which is between 1 and 20,
unit: second, 255-infinite*/
    "@min":0,
    "@max":20
},
"fingerPrintRecognizeInterval":{
/*optional, integer, fingerprint scanning interval, which is between 1 and 10,
unit: second, 255-no delay*/
    "@min":0,
    "@max":10
},
"fingerPrintMatchFastMode":{
/*optional, integer, fingerprint matching quick mode: 1-quick mode 1, 2-quick
mode 2, 3-quick mode 3, 4-quick mode 4, 5-quick mode 5, 255-automatic*/
    "@min":0,
    "@max":5
},
"fingerPrintModuleSensitive":{
/*optional, integer, fingerprint module sensitivity, which is between 1 and 8*/
    "@min":1,
    "@max":8
},
"fingerPrintModuleLightCondition":{
/*optional, string, fingerprint module light condition: "outdoor", "indoor"*/
    "@opt":"outdoor,indoor"
},
"faceMatchThresholdN":{
/*optional, integer, threshold of face picture 1:N comparison, which is between
0 and 100*/
    "@min":0,
    "@max":100
},
"faceQuality":{
/*optional, integer, face picture quality, which is between 0 and 100*/
    "@min":0,
    "@max":100
},
"faceRecognizeTimeOut":{
/*optional, integer, face recognition timeout, which is between 1 and 20, unit:
second, 255-infinite*/
    "@min":0,
    "@max":20
},
"faceRecognizeInterval":{
/*optional, integer, face recognition interval, which is between 1 and 10,
unit: second, 255-no delay*/
}

```

```
        "@min":0,
        "@max":10
    },
    "cardReaderFunction":{
/*ro, opt, array, card reader type: "fingerPrint"-fingerprint, "face",
"fingerVein"-finger vein*/
        "@opt":"fingerPrint,face,fingerVein"
    },
    "cardReaderDescription":{
/*ro, opt, card reader description. If the card reader is the Wiegand card
reader or if offline, this field will be set to "Wiegand" or "485Offline"*/
        "@min":1,
        "@max":16
    },
    "faceImageSensitometry":{
/*ro, opt, integer, face picture exposure, which is between 0 and 65535*/
        "@min":0,
        "@max":65535
    },
    "livingBodyDetect":"true,false",
/*optional, boolean, whether to enable human detection*/
    "faceMatchThreshold1":{
/*optional, integer, threshold of face picture 1:1 comparison, which is between
0 and 100*/
        "@min":0,
        "@max":100
    },
    "buzzerTime":{
/*optional, integer, buzzing duration, which is between 0 and 5999, unit:
second, 0-long buzzing*/
        "@min":0,
        "@max":5999
    },
    "faceMatch1SecurityLevel":{
/*optional, integer, security level of face 1:1 recognition: 1-normal, 2-high,
3-higher*/
        "@opt":"1,2,3"
    },
    "faceMatchNSecurityLevel":{
/*optional, integer, security level of face 1:N recognition: 1-normal, 2-high,
3-higher*/
        "@opt":"1,2,3"
    },
    "envirMode":{
/*optional, string, environment mode of face recognition: "indoor", "other"*/
        "@opt":"indoor,other"
    },
    "liveDetLevelSet":{
/*optional, string, threshold level of liveness detection: "low", "middle"-
medium, "high"*/
        "@opt":"low,middle,high"
    },
}
```

```

    "liveDetAntiAttackCntLimit": {
        /*optional, integer, number of anti-attacks of liveness detection, which is
        between 1 and 255. This value should be configured as the same one on both
        client and device*/
        "@min": 1,
        "@max": 255
    },
    "enableLiveDetAntiAttack": "true, false",
    /*optional, boolean, whether to enable anti-attack for liveness detection*/
    "supportDelFPByID": "true, false",
    /*ro, opt, boolean, whether the card reader supports deleting fingerprint by
    fingerprint ID: "true"-yes, "false"-no*/
    "fingerPrintCapacity": {
        /*ro, opt, integer, maximum number of fingerprints that can be added*/
        "@min": ,
        "@max": 
    },
    "fingerPrintNum": {
        /*ro, opt, integer, number of added fingerprints*/
        "@min": ,
        "@max": 
    },
    "defaultVerifyMode": {
        /*ro, opt, string, default authentication mode of the card reader (factory
        defaults): "cardAndPw"-card+password, "card", "cardOrPw"-card or password, "fp"-fingerprint,
        "fpAndPw"-fingerprint+password, "fpOrCard"-fingerprint or card,
        "fpAndCard"-fingerprint+card, "fpAndCardAndPw"-fingerprint+card+password,
        "faceOrFpOrCardOrPw"-face or fingerprint or card or password, "faceAndFp"-face
        +fingerprint, "faceAndPw"-face+password, "faceAndCard"-face+card, "face",
        "employeeNoAndPw"-employee No.+password, "fpOrPw"-fingerprint or password,
        "employeeNoAndFp"-employee No.+fingerprint, "employeeNoAndFpAndPw"-employee No.
        +fingerprint+password, "faceAndFpAndCard"-face+fingerprint+card,
        "faceAndPwAndFp"-face+password+fingerprint, "employeeNoAndFace"-employee No.
        +face, "faceOrfaceAndCard"-face or face+card, "fpOrface"-fingerprint or face,
        "cardOrfaceOrPw"-card or face or password, "cardOrFace"-card or face,
        "cardOrFaceOrFp"-card or face or fingerprint*/
        "@opt": "cardAndPw, card, cardOrPw, fp, fpAndPw, fpOrCard, fpAndCard, fpAndCardAndPw, fac
        eOrFpOrCardOrPw, faceAndFp, faceAndPw, faceAndCard, face, employeeNoAndPw, fpOrPw, empl
        oyeeNoAndFp, employeeNoAndFpAndPw, faceAndFpAndCard, faceAndPwAndFp, employeeNoAndFa
        ce, faceOrfaceAndCard, fpOrface, cardOrfaceOrPw, cardOrFace, cardOrFaceOrFp"
    },
    "faceRecognizeEnable": {
        /*optional, integer, whether to enable facial recognition: 1-enable, 2-disable,
        3-attendance checked in/out by recognition of multiple faces*/
        "@opt": "1, 2, 3"
    },
    "FPAlgorithmVersion": {
        /*optional, string, read-only, fingerprint algorithm library version*/
        "@min": ,
        "@max": 
    },
}

```

```

    "cardReaderVersion": {
        /*optional, string, read-only, card reader version*/
        "@min": ,
        "@max": ,
    }
    "enableReverseCardNo": "true,false",
    /*optional, boolean, whether to enable reversing the card No.*/
    "independSwipeIntervals": {
        /*optional, int, time interval of person authentication, unit: second. This
        time interval is calculated for each person separately and is different from
        swipeInterval*/
        "@min": ,
        "@max": ,
    },
    "maskFaceMatchThresholdN": {
        /*optional, int, 1:N face picture (face with mask and normal background)
        comparison threshold, value range: [0,100]*/
        "@min": ,
        "@max": ,
    },
    "maskFaceMatchThreshold1": {
        /*optional, int, 1:1 face picture (face with mask and normal background)
        comparison threshold, value range: [0,100]*/
        "@min":0,
        "@max":100
    }
}
}

```

## F.29 JSON\_Cap\_ClearAntiSneak

ClearAntiSneak capability message in JSON format

```

{
    "ClearAntiSneak": {
        "clearAll": "true,false",
        /*required, boolean, whether to clear all anti-passing back records: "true"-yes, "false"-no. Clearing all anti-passing back records is not supported by
        access control devices version 2.1*/
        "EmployeeNoList" : {
            /*optional, person ID list, this node is valid when clearAll is "false"*/
            "maxSize": ,
            "employeeNo": {
                /*optional, string, employee No. (person ID)*/
                "@min": ,
                "@max": ,
            }
        }
    }
}

```

## F.30 JSON\_Cap\_ClearAntiSneakCfg

ClearAntiSneakCfg capability message in JSON format

```
{
    "ClearAntiSneakCfg": {
        "ClearFlags": {
            "antiSneak": "true,false"
        }
    }
}
```

## F.31 JSON\_Cap\_ClearPlansCfg

ClearPlansCfg capability message in JSON format

```
{
    "ClearPlansCfg": {
        "ClearFlags": {
            "doorStatusWeekPlan": "true,false",
            /*optional, boolean, whether to clear the week schedule of the door control*/
            "cardReaderWeekPlan": "true,false",
            /*optional, boolean, whether to clear the week schedule of the card reader
            authentication mode control*/
            "userRightWeekPlan": "true,false",
            /*optional, boolean, whether to clear the week schedule of the access
            permission control*/
            "doorStatusHolidayPlan": "true,false",
            /*optional, boolean, whether to clear the holiday schedule of the door control*/
            "cardReaderHolidayPlan": "true,false",
            /*optional, boolean, whether to clear the holiday schedule of the card reader
            authentication mode control*/
            "userRightHolidayPlan": "true,false",
            /*optional, boolean, whether to clear the holiday schedule of the access
            permission control*/
            "doorStatusHolidayGroup": "true,false",
            /*optional, boolean, whether to clear the holiday group of the door control*/
            "cardReaderHolidayGroup": "true,false",
            /*optional, boolean, whether to clear the holiday group of the card reader
            authentication mode control*/
            "userRightHolidayGroup": "true,false",
            /*optional, boolean, whether to clear the holiday group of the access
            permission control*/
            "doorStatusTemplate": "true,false",
            /*optional, boolean, whether to clear the schedule template of the door
            control*/
            "cardReaderTemplate": "true,false",
            /*optional, boolean, whether to clear the control schedule template of the card
            reader*/
        }
    }
}
```

```
reader authentication mode*
    "userRightTemplate": "true,false"
/*optional, boolean, whether to clear the schedule template of the access
permission control*/
}
}
}
```

### F.32 JSON\_Cap\_EventOptimizationCfg

EventOptimizationCfg capability message in JSON format

```
{
  "EventOptimizationCfg":{
    "enable":"true,false",
/*optional, boolean, whether to enable event optimization: "true"-yes
(default), "false"-no*/
    "isCombinedLinkageEvents": "true,false"
/*optional, boolean, whether to enable linked event combination: "true"-yes
(default), "false"-no*/
  }
}
```

### F.33 JSON\_Cap\_FaceRecognizeMode

FaceRecognizeMode capability message in JSON format

```
{
  "FaceRecognizeMode":{
    "mode":{
/*optional, string type, facial recognition mode: "normalMode"-normal mode,
"deepMode"-deep mode/
      "@opt":"normalMode,deepMode"
    }
  }
}
```

### F.34 JSON\_Cap\_FingerPrintCfg

FingerPrintCfg capability message in JSON format

```
{
  "FingerPrintCfg":{
    "searchID":{
/*required, string type, search ID*/
      "@min":1,
      "@max":36
    }
  }
}
```

```

},
"employeeNo": {
/*required, string, employee No. (person ID) linked with the fingerprint*/
    "@min": ,
    "@max": 
},
"enableCardReader": {
/*required, array, fingerprint module to apply fingerprint data to*/
    "@min": ,
    "@max": 
},
"fingerPrintID": {
/*required, integer, fingerprint No., which is between 1 and 10*/
    "@min":1,
    "@max":10
},
"fingerType": {
/*required, string, fingerprint type: "normalFP"-normal fingerprint, "hijackFP"-duress fingerprint, "patrolFP"-patrol fingerprint, "superFP"-super fingerprint, "dismissingFP"-dismiss fingerprint*/
    "@opt":"normalFP,hijackFP,patrolFP,superFP,dismissingFP"
},
"leaderFP": {
/*optional, array, whether to support first time authentication function*/
    "@min":1,
    "@max":32
},
"checkEmployeeNo":"true,false",
/*optional, boolean, whether to judge the existence of the employee No. (person ID): "false"-no, "true"-yes. If this node is not configured, the device will judge the existence of the employee No. (person ID) by default. If this node is set to "false", the device will not judge the existence of the employee No. (person ID) to speed up data applying; if this node is set to "true" or NULL, the device will judge the existence of the employee No. (person ID), and it is recommended to set this node to "true" or NULL if there is no need to speed up data applying*/
    "callbackMode":"allRetrun,partReturn",
/*optional, string, device callback mode: "allRetrun"-return when applying to all fingerprint modules completed (blocking); "partReturn"-return when applying to a part of fingerprint modules completed (unblocking). If this node is set to NULL, blocking mode will be adopted*/
/*when blocking mode is adopted, the totalStatus must be 1 after FingerPrintStatus is returned, which indicates that the fingerprint information is applied; when unblocking mode is adopted, if the totalStatus is 0 after FingerPrintStatus is returned, you should repeatedly call the URI /ISAPI/AccessControl/FingerPrintProgress?format=json to get the applying progress (which is also returned in FingerPrintStatus) until totalStatus equals to 1 (the fingerprint information is applied)*/
    "StatusList": {
/*optional, status list*/
        "id": {
/*optional, integer, fingerprint module No.*/

```

```

        "@min": ,
        "@max":
    },
    "cardReaderRecvStatus":{
/*optional, integer, fingerprint module status: 0-connecting failed, 1-
connected, 2-the fingerprint module is offline, 3-the fingerprint quality is
poor, try again, 4-the memory is full, 5-the fingerprint already exists, 6-the
fingerprint ID already exists, 7-invalid fingerprint ID, 8-this fingerprint
module is already configured, 10-the fingerprint module version is too old to
support the employee No.*/
        "@opt": "0,1,2,3,4,5,6,7,8,10"
    },
    "errorMsg":{
/*optional, string, error information*/
        "@min": ,
        "@max":
    }
},
"totalStatus":{
/*required, integer, applying status: 0-applying, 1-applied*/
    "@opt":"0,1"
},
"isSupportFingerCover":true,
/*whether to support overwriting the original fingerprint when applying new
fingerprint linked with the same person ID or employee No. If it is supported,
this node will be set to "true"; otherwise, this node will not be returned*/
    "isSupportSetUp":true
/*whether to support setting fingerprint parameters. If it is supported, this
node will be set to "true"; otherwise, this node will not be returned*/
}
}

```

### F.35 JSON\_Cap\_FingerPrintDelete

FingerPrintDelete capability message in JSON format

```
{
    "FingerPrintDelete":{
        "mode":{
/*required, string, deleting mode: "byEmployeeNo"-delete by employee No.
(person ID), "byCardReader"-delete by fingerprint module*/
            "@opt":"byEmployeeNo,byCardReader"
        },
        "EmployeeNoDetail":{
/*optional, delete by employee No. (person ID), this node is valid when mode is
"byEmployeeNo"*/
            "employeeNo":{
/*optional, string, employee No. (person ID) linked with the fingerprint*/
                "@min": ,
                "@max":
            }
        }
    }
}
```

```

    },
    "enableCardReader":{
/*optional, array, fingerprint module whose fingerprints should be deleted*/
        "@min": ,
        "@max": 
    },
    "fingerPrintID":{
/*optional, array, No. of fingerprint to be deleted*/
        "@min": ,
        "@max": 
    },
    "CardReaderDetail":{
/*optional, delete by fingerprint module, this node is valid when mode is
"byCardReader"*/
        "cardReaderNo":{
/*optional, integer, fingerprint module No.*/
            "@min": ,
            "@max": 
        },
        "clearAllCard":"true,false",
/*optional, boolean, whether to delete the fingerprint information of all
cards: "false"-no (delete by employee No.), "true"-yes (delete the fingerprint
information of all employee No.)*/
        "employeeNo":{
/*optional, string, employee No. (person ID) linked with the fingerprint, this
node is valid when clearAllCard is "false"*/
            "@min": ,
            "@max": 
        }
    }
}
}

```

### F.36 JSON\_Cap\_HealthCodeCfg

JSON message about the configuration capability of the health code

```

{
    "enabled":{
/*required, object, whether to enable: true, false*/
        "@opt": [true, false]
    },
    "serverAddress":{
/*optional, object, address of the health code server. The value is a string,
which means that configuring IP address and port No. separately is not
supported*/
        "@min":1,
        "@max":128
    }
}

```

```
    }
}
```

### F.37 JSON\_Cap\_HealthCodeDisplayCfg

JSON message about the configuration capability of health code display

```
{
  "showHealthCode":{
/*required, object, whether to display the health code information: true,
false*/
    "@opt":[true, false]
  }
}
```

### F.38 JSON\_Cap\_LogModeCfg

LogModeCfg capability message in JSON format

```
{
  "LogModeCfg":{
    "type":{
/*optional, integer, log mode: 1-16 bytes (the host log can be stored by 25w,
and the employee No. can be stored by 16 bytes), 2-12 bytes (the host log can
be stored by 25w, and the employee No. can be stored by 12 bytes). This node
will be set to 1 by default*/
      "@opt":"1,2"
    }
  }
}
```

### F.39 JSON\_Cap\_OSDPModify

OSDPModify capability message in JSON format

```
{
  "OSDPModify":{
    "id":{
/*required, integer, range of the original OSDP card reader ID*/
      "@min": ,
      "@max": ,
      "newID":{
/*required, integer, new ID of the OSDP card reader*/
        "@min": ,
        "@max": ,
      }
    }
}
```

```
    }
}
```

### F.40 JSON\_Cap OSDPStatus

OSDPStatus capability message in JSON format

```
{
  "OSDPStatus": {
    "id": {
      /*required, integer, range of the OSDP card reader ID*/
      "@min": ,
      "@max": ,
      "status": "online,offline"
      /*required, string, online status: "online", "offline"*/
    }
  }
}
```

### F.41 JSON\_Cap\_RegionCalibrationCfg

JSON message about the calibration configuration capability of the temperature measurement area

```
{
  "enabled": {
    /*required, object, whether to enable the calibration: true, false*/
    "@opt": [true, false]
  },
  "FaceFrameCoordinate": {
    /*optional, object, face frame coordinate, the value is normalized to a number between 0 and 1000*/
    "height": {
      /*optional, object, height*/
      "@min": 0,
      "@max": 1000
    },
    "width": {
      /*optional, object, width*/
      "@min": 0,
      "@max": 1000
    },
    "x": {
      /*optional, object, x-coordinate*/
      "@min": 0,
      "@max": 1000
    },
    "y": {
      /*optional, object, y-coordinate*/
      "@min": 0,
      "@max": 1000
    }
  }
}
```

```

/*optional, object, Y-coordinate*/
    "@min":0,
    "@max":1000
}
}
}

```

## F.42 JSON\_Cap\_RegionCoordinate

JSON message about the configuration capability of the temperature measurement area

```

{
  "uniqueItems":{

/*required, object, range of the number of vertexes of the polygon, read-only*/
    "@min":3,
    "@max":10
  },
  "RegionCoordinate":{

/*optional, object, coordinate of the vertexes of the polygon, read-only*/
    "x":{

/*optional, object, X-coordinate, read-only*/
      "@min":0,
      "@max":1000
    },
    "y":{

/*optional, object, Y-coordinate, read-only*/
      "@min":0,
      "@max":1000
    }
  }
}

```

## F.43 JSON\_Cap\_RemoteCheck

Message about the capability of verifying the access control event remotely in JSON format.

```

{
  "RemoteCheck":{

    "serialNo":{

/*required, int, event serial No. which should be the same as that in the event
information message for uploading*/
      "@min":1,
      "@max":2000000000
    },
    "checkResult":{

/*required, string, verification result: "success"-verified, "failed"-verification failed*/
      "@opt":["success", "failed"]
    },
  }
}

```

```
"info":{  
/*optional, string, additional information*/  
    "@min":1,  
    "@max":  
    }  
}  
}
```

### F.44 JSON\_Cap\_RemoteControlBuzzer

RemoteControlBuzzer capability message in JSON format

```
{  
    "RemoteControlBuzzer":{  
        "cardReaderNo":{  
/*optional, integer, card reader No. (buzzer No.)*/  
            "@min": ,  
            "@max":  
        },  
        "cmd":{  
/*required, string, command: "start"-start buzzing, "stop"-stop buzzing*/  
            "@opt":"start,stop"  
        }  
    }  
}
```

### F.45 JSON\_Cap\_TemperatureMeasurementCfg

JSON message about the configuration capability of the temperature measurement parameters

```
{  
    "showTemperatureInfo":{  
/*optional, object, whether to display the temperature information: true,  
false*/  
        "@opt": [true, false]  
    },  
    "saveThermalPicture":{  
/*optional, object, whether to save the thermal picture: true, false*/  
        "@opt": [true, false]  
    },  
    "uploadThermalPicture":{  
/*optional, object, whether to upload the thermal picture: true, false*/  
        "@opt": [true, false]  
    },  
    "lowTemperatureEnabled":{  
/*optional, boolean, whether to enable temperature measurement in the low-  
temperature environment: true, false. When this function is enabled, if the  
face temperature is lower than 36 °C, the measured temperature will be mapped  
to that higher than 36 °C; temperatures higher than 36 °C will not be mapped*/  
    }  
}
```

```

        "@opt": [true, false]
    }
}

```

## F.46 JSON\_Cap\_UserInfo

UserInfo capability message in JSON format

```

{
    "UserInfo": {
        "supportFunction": {
            /*required, supported function of adding, deleting, editing, searching for
            person information, and getting total number of the added persons: "post"-add,
            "delete", "put"-edit, "get"-search, "setUp"-set*/
            "@opt": "post,delete,put,get,setUp"
        },
        "UserInfoSearchCond": {
            /*optional, search conditions*/
            "searchID": {
                /*required, string type, search ID, which is used to check the upper-level
                platform or system. If the platform or the system is the same one during two
                searching, the search history will be saved in the memory to speed up next
                searching*/
                "@min": 1,
                "@max": 36
            },
            "maxResults": {
                /*required, integer32, maximum number of search results*/
                "@min": 1,
                "@max": 30
            },
            "EmployeeNoList": {
                /*optional, person ID list*/
                "maxSize": 56,
                "employeeNo": {
                    /*optional, string, employee No. (person ID)*/
                    "@min": ,
                    "@max":
                }
            },
            "fuzzySearch": {
                /*optional, string, keywords for fuzzy search*/
                "@min": ,
                "@max":
            },
            "isSupportNumOfFace": 0,
            /*optional, integer, whether it supports number of linked face pictures when
            searching. If this field is not returned, it indicates that this function is
            not supported*/
            "isSupportNumOfFP": 0,
        }
    }
}

```

```

/*optional, integer, whether it supports number of linked fingerprints when
searching. If this field is not returned, it indicates that this function is
not supported*/
    "isSupportNumOfCard":0,
/*optional, integer, whether it supports number of linked cards when searching.
If this field is not returned, it indicates that this function is not
supported*/
    "groupIdList":{
/*optional, object, range of the department No. of local time and attendance*/
        "@size":1,
/*required, int, the maximum number of lists supported by the device*/
        "@min":1,
/*required, int, the minimum value among elements of the array*/
        "@max":1
/*required, int, the maximum value among elements of the array*/
    },
    "arrangeType":{
/*optional, object, shift schedule type: "" (shift schedule by individual).
Currently only the shift schedule by individual is supported. If this node
exists, it indicates searching for all persons with shift schedule by
individual*/
        "@opt":["personal"]
    }
},
    "UserInfoDelCond":{
/*optional, deleting conditions*/
        "EmployeeNoList":{
/*optional, person ID list (if this node does not exist, it indicates deleting
all person information)*/
            "maxSize":56,
            "employeeNo":{
/*optional, string, employee No. (person ID)*/
                "@min": ,
                "@max":
            }
        }
},
    "employeeNo":{
/*required, string, employee No. (person ID)*/
        "@min": ,
        "@max":
    },
    "name":{
/*optional, string, name*/
        "@min":1,
        "@max":32
    },
    "userType":{
/*required, string, person type: "normal"-normal person (household), "visitor",
"blackList"-person in blocklist*/
        "@opt":"normal,visitor,blackList"
    },
}

```

```

    "closeDelayEnabled":"true,false",
/*optional, boolean, whether to enable door close delay: "true"-yes, "false"-no*/
    "Valid": {
/*required, parameters of the effective period*/
        "enable":"true, false",
/*required, boolean, whether to enable the effective period: "false"-disable,
"true"-enable. If this node is set to "false", the effective period is
permanent*/
        "beginTime": {
/*required, start time of the effective period (if timeType does not exist or
is "local", the beginTime is the device local time, e.g.,: 2017-08-01T17:30:08;
if timeType is "UTC", the beginTime is UTC time, e.g.,:
2017-08-01T17:30:08+08:00)*/
            "@min":1,
            "@max":32
        },
        "endTime": {
/*required, end time of the effective period (if timeType does not exist or is
"local", the endTime is the device local time, e.g.,: 2017-08-01T17:30:08; if
timeType is "UTC", the endTime is UTC time, e.g.,: 2017-08-01T17:30:08+08:00)*/
            "@min":1,
            "@max":32
        },
        "timeRangeBegin":"",
/*optional, string, start time that can be configured for beginTime. If the
device does not return this node, the default start time that can be configured
for beginTime is "1970-01-01T00:00:00"*/
        "timeRangeEnd":"",
/*optional, string, end time that can be configured for endTime. If the device
does not return this node, the default end time that can be configured for
endTime is "2037-12-31T23:59:59"*/
        "timeType": {
/*optional, string, time type: "local"- device local time, "UTC"- UTC time*/
            "@opt": "local,UTC"
        }
    },
    "maxBelongGroup":4,
/*optional, integer, maximum number of groups that a person can belong to*/
    "belongGroup": {
/*optional, string, group*/
        "@min":1,
        "@max":32
    },
    "password": {
/*optional, string, password*/
        "@min":1,
        "@max":32
    },
    "doorRight": {
/*optional, string, No. of door or lock that has access permission*/
        "@min":1,

```

```

        "@max":32
    },
    "RightPlan":{
/*optional, door permission schedule (lock permission schedule)*/
        "maxSize":32,
        "doorNo":{
/*optional, integer, door No. (lock ID)*/
            "@min":1,
            "@max":32
        },
        "maxPlanTemplate":4,
/*optional, integer, maximum number of schedule templates that can be
configured for one door*/
        "planTemplateNo":{
/*optional, string, schedule template No.*/
            "@min":1,
            "@max":32
        }
    },
    "maxOpenDoorTime":{
/*optional, integer, maximum authentication attempts, 0-unlimited*/
        "@min":0,
        "@max":100
    },
    "openDoorTime":{
/*optional, integer, read-only, authenticated attempts*/
        "@min":0,
        "@max":100
    },
    "roomNumber":{
/*optional, integer, room No.*/
        "@min":0,
        "@max":100
    },
    "floorNumber":{
/*optional, integer, floor No.*/
        "@min":0,
        "@max":100
    },
    "doubleLockRight":"true, false",
/*optional, boolean, whether to have the permission to open the double-locked
door: "true"-yes, "false"-no*/
    "localUIRight":"true, false",
/*optional, boolean, whether to have the permission to access the device local
UI: "true"-yes, "false"-no*/
    "localUIUserType":{
/*optional, object, user type of device local UI: "admin" (administrator),
"operator", "viewer" (normal user). This node is used to distinguish different
users with different operation permissions of device local UI*/
        "@opt":["admin","operator","viewer"]
    },
    "userVerifyMode":{


```

```

/*optional, string, person authentication mode: "cardAndPw"-card+password,
"card"-card, "cardOrPw"-card or password, "fp"-fingerprint, "fpAndPw"-
fingerprint+password, "fpOrCard"-fingerprint or card, "fpAndCard"-fingerprint
+card, "fpAndCardAndPw"-fingerprint+card+password, "faceOrFpOrCardOrPw"-face or
fingerprint or card or password, "faceAndFp"-face+fingerprint, "faceAndPw"-face
+password, "faceAndCard"-face+card, "face"-face, "employeeNoAndPw"-employee No.
+password, "fpOrPw"-fingerprint or password, "employeeNoAndFp"-employee No.
+fingerprint, "employeeNoAndFpAndPw"-employee No.+fingerprint+password,
"faceAndFpAndCard"-face+fingerprint+card, "faceAndPwAndFp"-face+password
+fingerprint, "employeeNoAndFace"-employee No.+face, "faceOrfaceAndCard"-face
or face+card, "fpOrface"-fingerprint or face, "cardOrfaceOrPw"-card or face or
password, "cardOrFace"-card or face, "cardOrFaceOrFp"-card or face or
fingerprint, "cardOrFpOrPw"-card or fingerprint or password. The priority of
the person authentication mode is higher than that of the card reader
authentication mode*/
    },
    "checkUser":"true, false",
/*optional, boolean, whether to verify the duplicated person information:
"false"-no, "true"-yes. If checkUser is not configured, the device will verify
the duplicated person information by default. When there is no person
information, you can set checkUser to "false" to speed up data applying;
otherwise, it is not recommended to configure this node*/
    "addUser": "true, false",
/*optional, boolean type, whether to add the person if the person information
being edited does not exist: "false"-no (if the device has checked that the
person information being edited does not exist, the failure response message
will be returned along with the error code), "true"-yes (if the device has
checked that the person information being edited does not exist, the success
response message will be returned, and the person will be added). If this node
is not configured, the person will not be added by default*/
    "maxRecordNum": ,
/*required, integer type, supported maximum number of records (person records)*/
    "callNumbers": {
/*optional, string type, room No. list to be called, which is extended from
roomNumber and it is in higher priority; by default, the No. format is X-X-X-X,
e.g., 1-1-1-401, and for standard SIP, it can be the SIP number; this node must
be configured together with roomNumber*/
        "maxSize": ,
/*range of members in the array*/
        "@min": 0,
/*minimum string length*/
        "@max": 100
/*maximum string length*/
    },
    "floorNumbers": {
/*optional, integer type, floor No. list, which is extended from floorNumber

```

```

and it is in higher priority; this node must be configured together with
floorNumber/*
    "maxSize": ,
/*range of members in the array*/
    "@min": 0,
/*minimum floor No.*/
    "@max": 100
/*maximum floor No.*/
},
"gender":{
/*optional, string, gender of the person in the face picture: "male", "female",
"unknown"*/
    "@opt":"male,female,unknown"
},
"PersonInfoExtends": {
/*optional, object, extended fields for the additional person information*/
    "maxSize":3,
/*required, integer, supported maximum number of extension fields*/
    "id":{
/*optional, object, extended ID of the additional person information*/
        "@min": 1,
        "@max": 1
    },
    "value":{
/*optional, object, extended content of the additional person information*/
        "@min": 0,
        "@max": 100
    }
},
"purePwdVerifyEnable": ,
/*optional, boolean, whether the device supports opening the door only by
password: true-yes, this node is not returned-no. The password used to open the
door is the value of the node password in the message JSON_UserInfo.*/
/*For opening the door only by password: 1. The password in "XXX or password"
in the authentication mode refers to the person's password (the value of the
node password in JSON_UserInfo); 2. The device will not check the duplication
of the password, and the upper platform should ensure that the password is
unique; 3. The password cannot be added, deleted, edited, or searched for on
the device locally.*/
    "groupId":{
/*optional, object, department No. of local time and attendance*/
        "@min": 0,
        "@max": 0
    },
    "localAtndPlanTemplateId":{
/*optional, object, schedule template of local time and attendance. If this
node exist, it indicates that there are shift schedule settings by individual*/
        "@min": 0,
        "@max": 0
    }
}
}

```

## F.47 JSON\_Cap\_UserInfoDetail

UserInfoDetail capability message in JSON format

```
{
  "UserInfoDetail": {
    "mode": {
      "@opt": "all,byEmployeeNo"
      /*required, string type, deleting mode: "all"-delete all, "byEmployeeNo"-delete
      by employee No. (person ID)*/
    },
    "EmployeeNoList": {
      /*optional, person ID list*/
      "maxSize": ,
      "employeeNo": {
        /*optional, string type, employee No. (person ID), it is valid when mode is
        "byEmployeeNo"*/
        "@min": ,
        "@max": ,
      }
    }
  }
}
```

## F.48 JSON\_Cap\_UserRightHolidayGroupCfg

UserRightHolidayGroupCfg capability message in JSON format

```
{
  "UserRightHolidayGroupCfg": {
    "groupNo": {
      /*holiday group No.*/
      "@min": 1,
      "@max": 16
    },
    "enable": "true,false",
    /*whether to enable: "true"-enable, "false"-disable*/
    "groupName": {
      /*holiday group name*/
      "@min": 1,
      "@max": 32
    },
    "holidayPlanNo": {
      /*holiday group schedule No.*/
      "@min": 1,
      "@max": 16
    }
  }
}
```

```

    }
}
```

## F.49 JSON\_Cap\_UserRightHolidayPlanCfg

UserRightHolidayPlanCfg capability message in JSON format

```
{
  "UserRightHolidayPlanCfg": {
    "planNo": {
      /*holiday schedule No.*/
      "@min": 1,
      "@max": 16
    },
    "enable": "true,false",
    /*whether to enable: "true"-enable, "false"-disable*/
    "beginDate": "",
    /*start date of the holiday (device local time)*/
    "endDate": "",
    /*end date of the holiday (device local time)*/
    "HolidayPlanCfg": {
      /*holiday schedule parameter*/
      "maxSize": 8,
      "id": {
        /*time period No.*/
        "@min": 1,
        "@max": 8
      },
      "enable": "true,false",
      /*whether to enable: "true"-enable, "false"-disable*/
      "TimeSegment": {
        "beginTime": "",
        /*start time of the time period (device local time)*/
        "endTime": "",
        /*end time of the time period (device local time)*/
        "validUnit":
        /*time accuracy: "hour", "minute", "second". If this node is not returned, it
        indicates that the time accuracy is "minute"*/
      }
    }
  }
}
```

## F.50 JSON\_Cap\_UserRightPlanTemplate

UserRightPlanTemplate capability message in JSON format

```
{
  "UserRightPlanTemplate": {
```

```

    "templateNo": {
/*schedule template No.*/
        "@min": 1,
        "@max": 16
    },
    "enable": "true,false",
/*whether to enable: "true"-enable, "false"-disable*/
    "templateName": {
/*template name*/
        "@min": 1,
        "@max": 32
    },
    "weekPlanNo" : {
/*week schedule No.*/
        "@min": 1,
        "@max": 16
    },
    "holidayGroupNo": {
/*holiday group No.*/
        "@min": 1,
        "@max": 16
    }
}
}

```

### F.51 JSON\_Cap\_UserRightWeekPlanCfg

UserRightWeekPlanCfg capability message in JSON format

```

{
    "UserRightWeekPlanCfg": {
        "planNo": {
/*week schedule No.*/
            "@min":1,
            "@max":16
        },
        "enable": "true,false",
/*whether to enable: "true"-enable, "false"-disable*/
        "WeekPlanCfg": {
/*week schedule parameter*/
            "maxSize":56,
            "week": {
                "@opt": "Monday,Tuesday,Wednesday,Thursday,Friday,Saturday,Sunday"
            },
            "id": {
                "@min":1,
                "@max":8
            },
            "enable": "true,false",
/*whether to enable: "true"-enable, "false"-disable*/
        }
    }
}

```

```
"TimeSegment":{  
    "beginTime": "",  
    /*start time of the time period (device local time)*/  
    "endTime": "",  
    /*end time of the time period (device local time)*/  
    "validUnit":  
        /*time accuracy: "hour", "minute", "second". If this node is not returned, it  
        indicates that the time accuracy is "minute"*/  
    }  
}  
}
```

### F.52 JSON\_CapturePreset

CapturePreset message in JSON format

```
{  
    "CapturePreset":{  
        "name": ""  
        /*optional, string, name, the maximum size is 128 bytes by default. This field  
        is NULL by default*/  
    }  
}
```

### F.53 JSON\_CapturePresetCap

CapturePresetCap capability message in JSON format

```
{  
    "CapturePresetCap":{  
        "name": {  
            /*optional, string, name*/  
            "@min": 0,  
            "@max": 0  
        }  
    }  
}
```

### F.54 JSON\_CaptureProgress

CaptureProgress message in JSON format

```
{  
    "CaptureProgress":{  
        "reqCaptureNum": ,  
        /*optional, integer, total number of person to be collected*/  
    }  
}
```

```

    "completelyCaptureNum": ,
/*optional, integer, number of completely collected persons*/
    "partiallyCaptureNum": ,
/*optional, integer, number of partially collected persons*/
    "reqFaceNum": ,
/*optional, integer, number of faces to be collected*/
    "faceNum": ,
/*optional, integer, number of collected faces*/
    "reqFingerprintNum": ,
/*optional, integer, number of fingerprints to be collected*/
    "fingerprintNum": ,
/*optional, integer, number of collected fingerprints*/
    "reqCardNum": ,
/*optional, integer, number of cards to be collected*/
    "cardNum": ,
/*optional, integer, number of collected cards*/
    "reqIDCardNum": ,
/*optional, integer, number of ID cards to be collected*/
    "IDCardNum": ,
/*optional, integer, number of collected ID cards*/
    "reqIssueNum": ,
/*optional, int, number of persons to be issued with smart cards*/
    "IssuedNum": ,
/*optional, int, number of persons that have been issued with smart cards*/
}
}

```

### F.55 JSON\_CaptureRule

CaptureRule message in JSON format

```

{
  "CaptureRule": {
    "enableCardNoLenAuto": ,
/*optional, boolean, whether to enable length self-adaption of the card serial
No.*/
    "cardNoLen": ,
/*dependency, integer, length of the card serial No.: 3, 4, 7, 10, unit: byte.
This field is valid when enableCardNoLenAuto is "false". If this field is set
to 3, it refers to Wiegand 26*/
    "cardTimeout": ,
/*optional, integer, card collection timeout, unit: ms*/
  }
}

```

### F.56 JSON\_CaptureRuleCap

CaptureRuleCap capability message in JSON format

```
{
    "CaptureRuleCap": {
        "enableCardNoLenAuto": [true, false],
        /*optional, boolean, whether to enable length self-adaption of the card serial
        No.*/
        "cardNoLen": {
            /*dependency, integer, length of the card serial No.: 3, 4, 7, 10*/
            "@opt": [3, 4, 7, 10]
        },
        "cardTimeout": {
            /*optional, integer, card collection timeout, unit: ms*/
            "@min": 0,
            "@max": 0
        }
    }
}
```

### **F.57 JSON\_CardEncryption**

JSON message about card encryption parameters

```
{
    "CardEncryption": {
        "cardType": "",
        /*required, string type, card types: "blank"-blank card, "private"-private CPU
        card, encrypted-other encrypted cards*/
        "keyLen": ,
        /*depend, integer, size of key for external authentication, this field is valid
        only when cardType is set to "encrypted"*/
        "key": ""
        /*required, hexadecimal string, a 16-byte key content for external
        authentication*/
    }
}
```

### **F.58 JSON\_CardInfo**

JSON message about card information

```
{
    "CardInfo": {
        "employeeNo": "",
        /*required, string, employee No. (person ID)*/
        "cardNo": "",
        /*required, string, card No.*/
        "deleteCard": ,
        /*optional, boolean, whether to delete the card: "true"-yes. This node is
        required only when the card needs to be deleted; for adding or editing card*/
    }
}
```

```

information, this node can be set to NULL*/
    "cardType":"",
/*optional, string, card type: "normalCard"-normal card, "patrolCard"-patrol
card, "hijackCard"-duress card, "superCard"-super card, "dismissingCard"-dismiss
card, "emergencyCard"-emergency card (it is used to assign permission to a temporary
card, but it cannot open the door)*/
    "leaderCard":"",
/*optional, string, whether to support first card authentication function,
e.g., the value "1,3,5" indicates that the access control points No.1, No.3,
and No.5 support first card authentication function*/
    "checkCardNo":"",
/*optional, boolean, whether to enable duplicated card verification: "false"-disable,
"true"-enable. If this node is not configured, the device will verify the duplicated
card by default. When there is no card information, you can set checkCardNo to "false"
to speed up data applying; otherwise, it is not recommended to configure this node*/
    "checkEmployeeNo": ,
/*optional, boolean, whether to check the existence of the employee No. (person
ID): "false"-no, "true"-yes. If this node is not configured, the device will
check the existence of the employee No. (person ID) by default. If this node is
set to "false", the device will not check the existence of the employee No.
(person ID) to speed up data applying; if this node is set to "true" or NULL,
the device will check the existence of the employee No. (person ID), and it is
recommended to set this node to "true" or NULL if there is no need to speed up
data applying*/
    "addCard": ,
/*optional, boolean, whether to add the card if the card information being
edited does not exist: "false"-no (if the device has checked that the card
information being edited does not exist, the failure response message will be
returned along with the error code), "true"-yes (if the device has checked that
the card information being edited does not exist, the success response message
will be returned, and the card will be added). If this node is not configured,
the card will not be added by default*/
    "operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal*/
    "terminalNoList": [1]
/*optional, array, terminal ID list, this node is required when operation type
is "byTerminal"; currently, only one terminal is supported*/
}
}

```

### Remarks

The **employeeNo** and **cardNo** cannot be edited. If you need to edit the **cardNo**, you should delete the previous card and create a new card.

## F.59 JSON\_CardInfo\_Collection

CardInfo message in JSON format

```
{
  "CardInfo": {
    "cardNo": "",
    /*required, string, card No.*/
    "cardType": ""
    /*optional, string, card type: "TypeA_M1", "TypeA_CPU", "TypeB", "ID_125K",
    "FelicaCard"-Felica card, "DesfireCard"-DESFire card*/
  }
}
```

## F.60 JSON\_CardInfoCap

CardInfoCap capability message in JSON format

```
{
  "CardInfoCap": {
    "cardNo": {
      /*required, string, card No.*/
      "@min": 1,
      "@max": 32
    },
    "cardType": [
      "TypeA_M1", "TypeA_CPU", "TypeB", "ID_125K", "FelicaCard", "DesfireCard"
    ]
    /*optional, string, card type: "TypeA_M1", "TypeA_CPU", "TypeB", "ID_125K",
    "FelicaCard"-Felica card, "DesfireCard"-DESFire card*/
  }
}
```

## F.61 JSON\_CardInfoCount

CardInfoCount message in JSON format

```
{
  "CardInfoCount": {
    "cardNumber": /*number of cards*/
  }
}
```

## F.62 JSON\_CardInfoDelCond

JSON message about card information to be deleted

```
{
  "CardInfoDelCond": {
    "EmployeeNoList" : [ {
```

```

/*optional, person ID list, if this node does not exist or is set to NULL, it
indicates deleting all cards*/
    "employeeNo":"",
/*optional, string, employee No. (person ID)*/
    ],
    "CardNoList":{},
/*optional, card No. list (this node cannot exist together with the
EmployeeNoList, and if this node does not exist or is set to NULL, it indicates
deleting all cards)*/
    "cardNo":"",
/*optional, string, card No.*/
    ],
    "operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal*/
    "terminalNoList": [1]
/*optional, array, terminal ID list, this node is required when operation type
is "byTerminal"; currently, only one terminal is supported*/
}
}

```

## F.63 JSON\_CardInfoSearch

CardInfoSearch message in JSON format

```

{
    "CardInfoSearch": {
        "searchID": "",
/*required, string, search ID, which is used to confirm the upper-level
platform or system. If the platform or the system is the same one during two
searching, the search history will be saved in the memory to speed up next
searching*/
        "responseStatusStrg": "",
/*required, string, search status: "OK"-searching completed, "NO MATCH"-no
matched results, "MORE"-searching for more results*/
        "numOfMatches": ,
/*required, integer32, number of returned results*/
        "totalMatches": ,
/*required, integer32, total number of matched results*/
        "CardInfo": [
/*optional, person information*/
            "employeeNo": "",
/*required, string, employee No. (person ID)*/
            "cardNo": "",
/*required, string, card No.*/
            "cardType": "",
/*required, string, card type: "normalCard"-normal card, "patrolCard"-patrol
card, "hijackCard"-duress card, "superCard"-super card, "dismissingCard"-'
dismiss card, "emergencyCard"-emergency card (it is used to assign permission
to a temporary card, but it cannot open the door)*/
            "leaderCard": ""
        ]
    }
}

```

```

/*optional, string, whether to support first card authentication function,
e.g., the value "1,3,5" indicates that the access control points No.1, No.3,
and No.5 support first card authentication function*/
    }]
}
}

```

### F.64 JSON\_CardInfoSearchCond

CardInfoSearchCond message in JSON format

```

{
  "CardInfoSearchCond": {
    "searchID": "",

    /*required, string, search ID, which is used to confirm the upper-level
    platform or system. If the platform or the system is the same one during two
    searching, the search history will be saved in the memory to speed up next
    searching*/
    "searchResultPosition": ,

    /*required, integer32, the start position of the search result in the result
    list. When there are multiple records and you cannot get all search results at
    a time, you can search for the records after the specified position next time.
    For example, if the maximum total number of matched results (totalMatches)
    supported by the device is M and the total number of matched results
    (totalMatches) stored in the device currently is N (here N is smaller than M),
    the valid range of this field is from 0 to N-1*/
    "maxResults": ,

    /*required, integer32, maximum number of search results. If maxResults exceeds
    the range returned by the device capability, the device will return the maximum
    number of search results according to the device capability and will not return
    error message*/
    "EmployeeNoList": [{

      /*optional, person ID list (if this node does not exist or is set to NULL, it
      indicates searching for all cards)*/
      "employeeNo": ""

      /*optional, string, employee No. (person ID)*/
      }],


      "CardNoList": [{

        /*optional, card No. list (this node cannot exist together with EmployeeNoList,
        and if this node does not exist or is set to NULL, it indicates searching for
        all cards)*/
        "cardNo": ""

        /*optional, string, card No.*/
        }]

    }
}

```

## F.65 JSON\_CardOperationsCap

JSON message about card operation capability

```
{  
    "CardOperationsCap":{  
        "SectionEncryption":{  
            "supportFunction":{  
/*required, string, supported methods*/  
                "@opt": ["put", "get", "delete", "post"]  
            },  
            "sectionNo":{  
/*required, integer, section No.*/  
                "@min": 0,  
                "@max": 0  
            },  
            "keyType":{  
/*required, string, verification key types: "private"-private key, "normal"-  
other valid keys*/  
                "@opt": ["private", "normal"]  
            },  
            "password":{  
/*optional, string, a hexadecimal verification key, this field is valid only  
when keyType is set to "nomal"*/  
                "@min": 0,  
                "@max": 0  
            },  
            "newKeyType":{  
/*required, string, new key types: "private"-private key, "normal"-other valid  
keys*/  
                "@opt": ["private", "normal"]  
            },  
            "KeyA":{  
/*optional, string, a hexadecimal key A password*/  
                "@min": 0,  
                "@max": 0  
            },  
            "KeyB":{  
/*optional, string, a hexadecimal key B password*/  
                "@min": 0,  
                "@max": 0  
            },  
            "controlBits":{  
/*optional, string, a hexadecimal control bit*/  
                "@min": 0,  
                "@max": 0  
            }  
        },  
        "Verification":{  
            "supportFunction":{  
/*required, string, supported methods*/  
        }
```

```

        "@opt": ["put", "get", "delete", "post"]
    },
    "sectionNo":{
/*required, integer, section No.*/
        "@min": 0,
        "@max": 0
    },
    "passwordType":{
/*optional, password types: "KeyA" (default), "KeyB"*/
        "@opt": ["KeyA", "KeyB"]
    },
    "password":{
/*optional, string, a hexadecimal password*/
        "@min": 0,
        "@max": 0
    }
},
    "DataBlock":{
        "supportFunction":{
/*required, string, supported methods*/
            "@opt": ["put", "get", "delete", "post"]
        },
        "addressOfBlock":{
/*optional, integer, block address*/
            "@min": 0,
            "@max": 0
        },
        "data":{
/*required, a hexBinary string, e.g., "f2345678abf2345678abf2345678abf2"*/
            "@min": 0,
            "@max": 0
        },
        "DataBlockCtrl":{
            "supportFunction":{
/*required, string, supported methods*/
                "@opt": ["put", "get", "delete", "post"]
            },
            "addressOfBlock":{
/*required, integer, block address*/
                "@min": 0,
                "@max": 0
            },
            "command":{
/*required, string, control commands*/
                "@opt": ["add", "minus", "copy", "paste"]
            },
            "value":{
/*depend, integer, relative value to be changed, this field is valid only when
the command is set to "add" or "minus"*/
                "@min": 0,
                "@max": 0
            }
        }
    }
}

```

```

    },
},
"ControlBlock": {
    "supportFunction": {
/*required, string, supported methods*/
        "@opt": ["put", "get", "delete", "post"]
    },
    "sectionNo": {
/*required, integer, section No.*/
        "@min": 0,
        "@max": 0
    },
    "KeyA": {
/*optional, string, a hexadecimal key A*/
        "@min": 0,
        "@max": 0
    },
    "KeyB": {
/*optional, string, a hexadecimal key B*/
        "@min": 0,
        "@max": 0
    },
    "controlBits": {
/*optional, string, a hexadecimal control bit*/
        "@min": 0,
        "@max": 0
    }
},
"CardProto": {
    "supportFunction": {
/*required, string, supported methods*/
        "@opt": ["put", "get", "delete", "post"]
    },
    "protocol": {
/*required, string, operation protocol types*/
        "@opt": ["TypeA", "TypeB", "TypeAB", "125K", "all"]
    }
},
"CardEncryption": {
    "supportFunction": {
/*required, string, supported methods*/
        "@opt": ["put", "get", "delete", "post"]
    },
    "cardType": {
/*required, string, card types: "blank"-blank card, "private"-private CPU card,
"encrypted"-other encrypted card*/
        "@opt": [ "blank", "private", "encrypted" ]
    }
},
"keyLen": {
/*depend, integer, size of key for external authentication, this field is valid
only when cardType is set to "encrypted"*/
    "@min": 0,
}

```

```
        "@max": 0
    },
    "key": {
/*required, hexadecimal string, a 16-byte key content for external
authentication*/
        "@min": 0,
        "@max": 0
    }
},
"CardParam": {
    "supportFunction": {
/*required, string, supported methods*/
        "@opt": ["put", "get", "delete", "post"]
    },
    "type": {
/*required, string, card types*/
        "@opt": ["CPU1356", "PSAM1", "PSAM2", "PSAM3", "PSAM4"]
    },
    "protocol": {
/*required, string, card protocol types*/
        "@opt": ["T0", "T1"]
    }
},
"CardResetResponse": {
    "supportFunction": {
/*required, string, supported methods*/
        "@opt": ["put", "get", "delete", "post"]
    },
    "data": {
/*required, string, resetting response information (usually, it is
manufacturer, which is encoded by Base64 and specified by device)*/
        "@min": 0,
        "@max": 0
    }
},
"DataTrans": {
    "supportFunction": {
/*required, string, supported methods*/
        "@opt": ["put", "get", "delete", "post"]
    },
    "content": {
/*required, string, data to be passed through, which is encoded in Base64*/
        "@min": 0,
        "@max": 0
    }
},
"Issue": {
/*capability of sending a request for card issuing and getting the current card
issuing status and real-time card issuing results, related URIs: /ISAPI/
AccessControl/CardOperations/localIssueRequest?format=json and /ISAPI/
AccessControl/CardOperations/localIssueStatus?format=json*/
    "supportFunction": {
```

```

/*required, string, supported methods. The actually supported methods will be
returned*/
    "@opt": ["put", "get", "delete", "post"]
},
"LocalIssueRequest": {
    "operation": {
/*required, string, operation type: "face"-issue card to be enrolled with face
picture, "fingerprint"-issue card to be enrolled with fingerprint*/
        "@opt": ["face", "fingerprint"]
    },
    "FPIIndex": {
/*optional, int, fingerprint storage index (card storage area). This field is
valid when operation is "fingerprint"*/
        "@min": 0,
        "@max": 0
    },
    "facePic": {
/*optional, string, face picture type: "visible"-visible light picture,
"infrared"-IR light picture. This field is valid when operation is "face"*/
        "@opt": ["visible", "infrared"]
    }
},
"LocalIssueRes": {
    "status": {
/*required, string, card issuing status: "ok"-succeeded, "failed"-card
operation failed, "timeout"-timed out, "verifiyFailure"-authentication failed,
"noCard"-no card detected, "processing"-processing*/
        "@opt": ["ok", "failed", "processing", "timeout", "verifiyFailure",
"noCard"]
    },
    "cardNo": {
/*optional, string, issued card No.*/
        "@min": 0
    },
    "cardErrorCode": {
/*dependent, string, internal error code of card operation returned by the
device*/
        "@opt": []
    }
},
"localIssueCfg": {
/*capability of configuring rule parameters for issuing smart cards, related
URI: /ISAPI/AccessControl/CardOperations/localIssueCfg?format=json*/
    "validFP": {
/*optional, array of int, valid fingerprint ID. This field is valid for
applying fingerprint to the card*/
        "@size": 2,
        "@min": 1,
        "@max": 10
    },
    "validFacePicture": {

```

```
/*optional, string, valid face picture type: "visible"-visible light picture,
"infrared"-IR light picture. This field is valid for applying face picture to
the card*/
    "@opt":["visible", "infrared"]
}
},
"ClearData":{

/*capability of deleting data from the card, related URI: /ISAPI/AccessControl/
CardOperations/clearData?format=json*/
    "supportFunction":{

/*required, string, supported methods. The actually supported methods will be
returned*/
        "@opt":["put", "get", "delete", "post"]
},
    "checkAll":{

/*optional, boolean, whether to delete all data*/
        "@opt":[true, false]
},
    "checkFingerprint":{

/*optional, boolean, whether to delete fingerprint data. This field is valid
when checkAll is false or does not exist*/
        "@opt":[true, false]
},
    "fingerprints":{

/*optional, array of int, list of addresses whether the fingerprints to be
deleted are stored. This field is valid when checkFingerprint exists. If this
field does not exist, it indicates deleting all fingerprints*/
        "@size":2,
        "@min":0,
        "@max":0
},
    "checkFacePicture":{

/*optional, boolean, whether to delete face data. This field is valid when
checkAll is false or does not exist*/
        "@opt":[true, false]
},
    "checkCustom":{

/*optional, boolean, whether to delete custom data. This field is valid when
checkAll is false or does not exist*/
        "@opt":[true, false]
},
    "ClearDataRes":{

/*required, string, card issuing status: "ok"-succeeded, "failed"-operation
failed, "timeout"-timed out, "verifiyFailure"-authentication failed, "noCard"-no
card detected, "processing"-processing*/
        "@opt":["ok", "failed", "processing", "timeout", "verifiyFailure",
"noCard"]
},
    "cardErrorCode":{

/*dependent, int, internal error code of card operation*/
        "@opt":
```

```

        }
    },
    "CustomData":{
/*capability of setting custom card information, related URI: /ISAPI/
AccessControl/CardOperations/customData?format=json*/
        "supportFunction":{
/*required, string, supported methods. The actually supported methods will be
returned*/
            "@opt":["put", "get", "delete", "post"]
        },
        "address":{
/*optional, int, start address for writing. By default the data will be written
from the start address*/
            "@min":0,
            "@max":0
        },
        "length":{
/*optional, int, length of source data to be written, it is 0 by default, unit:
byte*/
            "@min":0,
            "@max":0
        },
        "data":{
/*required, string, custom information encoded by Base64*/
            "@min":0,
            "@max":0
        },
        "CustomDataRes":{
            "status":{
/*required, string, card issuing status: "ok"-succeeded, "failed"-operation
failed, "timeout"-timed out, "verifiyFailure"-authentication failed, "noCard"-no
card detected, "processing"-processing*/
                "@opt":["ok", "failed", "processing", "timeout", "verifiyFailure",
"noCard"]
            },
            "cardErrorCode":{
/*dependent, int, internal error code of card operation*/
                "@opt":
            }
        },
        "CustomDataSearchCond":{
/*condition configuration capability of searching for custom card information,
related URI: /ISAPI/AccessControl/CardOperations/customData/searchTask?
format=json*/
            "address":{
/*optional, int, start address for reading. By default the data will be read
from the start address*/
                "@min":0,
                "@max":0
            },

```

```

        "length":{
/*optional, int, length of data to be read, it is 0 by default, unit: byte*/
            "@min":0,
            "@max":0
        },
        "CustomDataResult":{
/*result capability of searching for custom card information, related URI: /ISAPI/AccessControl/CardOperations/customData/searchTask?format=json*/
            "length":{
/*required, int, length of data that has been read, unit: byte*/
                "@min":0,
                "@max":0
            },
            "data":{
/*required, string, card information encoded by Base64*/
                "@min":0,
                "@max":0
            },
            "status":{
/*required, string, card issuing status: "ok"-succeeded, "failed"-operation failed, "timeout"-timed out, "verifiyFailure"-authentication failed, "noCard"-no card detected, "processing"-processing*/
                "@opt":["ok", "failed", "processing", "timeout", "verifiyFailure",
"noCard"]
            },
            "cardErrorCode":{
/*required, int, internal error code of card operation*/
                "@opt":
            }
        },
        "CardIssueStatus":{
/*capability of getting the smart card issuing status, related URI: /ISAPI/AccessControl/CardOperations/cardIssueStatus?format=json*/
            "status":{
/*required, string, card issuing status: "ok"-succeeded, "failed"-operation failed, "timeout"-timed out, "verifiyFailure"-authentication failed, "noCard"-no card detected, "processing"-processing*/
                "@opt":["ok", "failed", "processing", "timeout", "verifiyFailure",
"noCard"]
            },
            "cardNo":{
/*optional, string, issued card No.*/
                "@min":0,
                "@max":0
            },
            "cardErrorCode":{
/*dependent, int, internal error code of card operation*/
                "@opt":
            },
            "face":{
/*optional, boolean, issuing status of the card containing the face picture:*/

```

```
true-issued, false-not issued*/
    "@opt": [true, false]
},
"fingerprint1": {
/*optional, boolean, issuing status of the card containing fingerprint 1: true-
issued, false-not issued*/
    "@opt": [true, false]
},
"fingerprint2": {
/*optional, boolean, issuing status of the card containing fingerprint 2: true-
issued, false-not issued*/
    "@opt": [true, false]
},
"customData": {
/*optional, boolean, issuing status of the card containing custom information:
true-issued, false-not issued*/
    "@opt": [true, false]
}
}
```

## F.66 JSON\_CardParam

JSON message about card parameters

```
{
  "CardParam": {
    "type": ""
/*required, string, card types: "CPU1356,PSAM1,PSAM2,PSAM3,PSAM4"*/
    "protocol": ""
/*required, string, card protocol types: "T0,T1"*/
  }
}
```

## F.67 JSON\_CardProto

JSON message about operation protocol types of card

```
{
  "CardProto": {
    "protocol": "TypeA"
/*required, string, operation protocol types: "TypeA,TypeB,TypeAB,125K,all"*/
  }
}
```

## F.68 JSON\_CardReaderAntiSneakCfg

CardReaderAntiSneakCfg message in JSON format

```
{
  "CardReaderAntiSneakCfg": {
    "enable": ,
    /*required, boolean, whether to enable the anti-passing back function of the
    card reader: "true"-enable, "false"-disable*/
    "followUpCardReader": ,
    /*optional, array, following card reader No. after the first card reader, e.g.,
    [2,3,4] indicates that card reader No. 2, No. 3, and No. 4 can be swiped after
    the first card reader*/
  }
}
```

## F.69 JSON\_CardReaderCfg

CardReaderCfg message in JSON format

```
{
  "CardReaderCfg": {
    "enable": ,
    /*required, boolean, whether to enable: "true"-yes, "false"-no*/
    "okLedPolarity": "",
    /*optional, string, OK LED polarity: "cathode", "anode"*/
    "errorLedPolarity": "",
    /*optional, string, error LED polarity: "cathode", "anode"*/
    "buzzerPolarity": "",
    /*optional, string, buzzer polarity: "cathode", "anode"*/
    "swipeInterval": ,
    /*optional, integer, time interval of repeated authentication, which is valid
    for authentication modes such as fingerprint, card, face, etc., unit: second*/
    "pressTimeout": ,
    /*optional, integer, timeout to reset entry on keypad, unit: second*/
    "enableFailAlarm": ,
    /*optional, boolean, whether to enable excessive failed authentication attempts
    alarm*/
    "maxReadCardFailNum": ,
    /*optional, integer, maximum number of failed authentication attempts*/
    "enableTamperCheck": ,
    /*optional, boolean, whether to enable tampering detection*/
    "offlineCheckTime": ,
    /*optional, integer, time to detect after the card reader is offline, unit:
    second*/
    "fingerPrintCheckLevel": ,
    /*optional, integer, fingerprint recognition level: 1-1/10 false acceptance
    rate (FAR), 2-1/100 false acceptance rate (FAR), 3-1/1000 false acceptance rate
    (FAR), 4-1/10000 false acceptance rate (FAR), 5-1/100000 false acceptance rate
    (FAR)*/
```

```

(FAR), 6-1/1000000 false acceptance rate (FAR), 7-1/10000000 false acceptance
rate (FAR), 8-1/100000000 false acceptance rate (FAR), 9-3/100 false acceptance
rate (FAR), 10-3/1000 false acceptance rate (FAR), 11-3/10000 false acceptance
rate (FAR), 12-3/100000 false acceptance rate (FAR), 13-3/1000000 false
acceptance rate (FAR), 14-3/10000000 false acceptance rate (FAR),
15-3/100000000 false acceptance rate (FAR), 16-Automatic Normal, 17-Automatic
Secure, 18-Automatic More Secure (currently not support)*/
    "useLocalController": ,
/*ro, opt, boolean, whether it is connected to the distributed controller*/
    "localControllerID": ,
/*ro, opt, integer, distributed controller No., which is between 1 and 64, 0-
unregistered. This field is valid only when useLocalController is "true"*/
    "localControllerReaderID": ,
/*ro, opt, integer, card reader ID of the distributed controller, 0-
unregistered. This field is valid only when useLocalController is "true"*/
    "cardReaderChannel": ,
/*ro, opt, integer, communication channel No. of the card reader: 0-Wiegand or
offline, 1-RS-485A, 2-RS-485B. This field is valid only when useLocalController
is "true"*/
    "fingerPrintImageQuality": ,
/*opt, integer, fingerprint image quality: 1-low quality (V1), 2-medium quality
(V1), 3-high quality (V1), 4-highest quality (V1), 5-low quality (V2), 6-medium
quality (V2), 7-high quality (V2), 8-highest quality (V2)*/
    "fingerPrintContrastTimeOut": ,
/*optional, integer, fingerprint comparison timeout, which is between 1 and 20,
unit: second, 255-infinite*/
    "fingerPrintRecognizeInterval": ,
/*optional, integer, fingerprint scanning interval, which is between 1 and 10,
unit: second, 255-no delay*/
    "fingerPrintMatchFastMode": ,
/*optional, integer, fingerprint matching quick mode: 1-quick mode 1, 2-quick
mode 2, 3-quick mode 3, 4-quick mode 4, 5-quick mode 5, 255-automatic*/
    "fingerPrintModuleSensitive": ,
/*optional, integer, fingerprint module sensitivity, which is between 1 and 8*/
    "fingerPrintModuleLightCondition": "",
/*optional, string, fingerprint module light condition: "outdoor", "indoor"*/
    "faceMatchThresholdN": ,
/*optional, integer, threshold of face picture 1:N comparison, which is between
0 and 100*/
    "faceQuality": ,
/*optional, integer, face picture quality, which is between 0 and 100*/
    "faceRecognizeTimeOut": ,
/*optional, integer, face recognition timeout, which is between 1 and 20, unit:
second, 255-infinite*/
    "faceRecognizeInterval": ,
/*optional, integer, face recognition interval, which is between 1 and 10,
unit: second, 255-no delay*/
    "cardReaderFunction": ,
/*ro, opt, array, card reader type: "fingerPrint"-fingerprint, "face",
"fingerVein"-finger vein. For example, ["fingerPrint","face"] indicates that
the card reader supports both fingerprint and face*/
    "cardReaderDescription": ""

```

```
/*ro, opt, string, card reader description. If the card reader is the Wiegand  
card reader or if offline, this field will be set to "Wiegand" or "485Offline"*/  
    "faceImageSensitometry": ,  
/*ro, opt, integer, face picture exposure, which is between 0 and 65535*/  
    "livingBodyDetect": ,  
/*optional, boolean, whether to enable human detection*/  
    "faceMatchThreshold1": ,  
/*optional, integer, threshold of face picture 1:1 comparison, which is between  
0 and 100*/  
    "buzzerTime": ,  
/*optional, integer, buzzing duration, which is between 0 and 5999, unit:  
second, 0-long buzzing*/  
    "faceMatch1SecurityLevel": ,  
/*optional, integer, security level of face 1:1 recognition: 1-normal, 2-high,  
3-higher*/  
    "faceMatchNSecurityLevel": ,  
/*optional, integer, security level of face 1:N recognition: 1-normal, 2-high,  
3-higher*/  
    "envirMode":"" ,  
/*optional, string, environment mode of face recognition: "indoor", "other"*/  
    "liveDetLevelSet": "",  
/*optional, string, threshold level of liveness detection: "low", "middle"-  
medium, "high"*/  
    "liveDetAntiAttackCntLimit": ,  
/*optional, integer, number of anti-attacks of liveness detection, which is  
between 1 and 255. This value should be configured as the same one on both  
client and device*/  
    "enableLiveDetAntiAttack": ,  
/*optional, boolean, whether to enable anti-attack for liveness detection*/  
    "supportDelFPByID": ,  
/*ro, opt, boolean, whether the card reader supports deleting fingerprint by  
fingerprint ID: "true"-yes, "false"-no*/  
    "fingerPrintCapacity": ,  
/*ro, opt, integer, fingerprint capacity, which is the maximum number of  
fingerprints that can be added*/  
    "fingerPrintNum": ,  
/*ro, opt, integer, number of added fingerprints*/  
    "defaultVerifyMode": "",  
/*ro, opt, string, default authentication mode of the fingerprint and card  
reader (factory defaults): "cardAndPw"-card+password, "card", "cardOrPw"-card  
or password, "fp"-fingerprint, "fpAndPw"-fingerprint+password, "fpOrCard"-  
fingerprint or card, "fpAndCard"-fingerprint+card, "fpAndCardAndPw"-fingerprint  
+card+password, "faceOrFpOrCardOrPw"-face or fingerprint or card or password,  
"faceAndFp"-face+fingerprint, "faceAndPw"-face+password, "faceAndCard"-face  
+card, "face", "employeeNoAndPw"-employee No.+password, "fpOrPw"-fingerprint or  
password, "employeeNoAndFp"-employee No.+fingerprint, "employeeNoAndFpAndPw"-  
employee No.+fingerprint+password, "faceAndFpAndCard"-face+fingerprint+card,  
"faceAndPwAndFp"-face+password+fingerprint, "employeeNoAndFace"-employee No.  
+face, "faceOrfaceAndCard"-face or face+card, "fpOrface"-fingerprint or face,  
"cardOrfaceOrPw"-card or face or password, "cardOrFace"-card or face,  
"cardOrFaceOrFp"-card or face or fingerprint*/  
    "faceRecognizeEnable": ,
```

```
/*optional, integer, whether to enable facial recognition: 1-enable, 2-disable,  
3-attendance checked in/out by recognition of multiple faces*/  
    "FPAgorithmVersion": "",  
/*optional, string, read-only, fingerprint algorithm library version*/  
    "cardReaderVersion": "",  
/*optional, string, read-only, card reader version*/  
    "enableReverseCardNo": true,  
/*optional, boolean, whether to enable reversing the card No.*/  
    "independSwipeIntervals": 0,  
/*optional, int, time interval of person authentication, unit: second. This  
time interval is calculated for each person separately and is different from  
swipeInterval*/  
    "maskFaceMatchThresholdN": 1,  
/*optional, int, 1:N face picture (face with mask and normal background)  
comparison threshold, value range: [0,100]*/  
    "maskFaceMatchThreshold1": 1  
/*optional, int, 1:1 face picture (face with mask and normal background)  
comparison threshold, value range: [0,100]*/  
    }  
}
```

## F.70 JSON\_CardResetResponse

JSON message about card resetting response

```
{  
    "CardResetResponse": {  
        "data": ""  
    /*required, string, resetting response information (usually, it is  
manufacturer, which is encoded by Base64 and specified by device*/  
    }  
}
```

## F.71 JSON\_CardVerificationRule

JSON message about the parameters of card No. authentication mode

```
{  
    "CardVerificationRule": {  
        "cardNoLenMode": "full"  
    /*required, string, length mode of card No. authentication (comparison):  
"full", "3Bytes", "4Bytes". After the card No. authentication (comparison) mode  
is switched, the device should check the card No. compatibility*/  
    }  
}
```

## F.72 JSON\_CardVerificationRuleCap

JSON message about the configuration capability of the card No. authentication mode

```
{
    "CardVerificationRuleCap": {
        "cardNoLenMode": {
            /*required, string, length mode of card No. authentication (comparison):
            "full", "3Bytes", "4Bytes". After the card No. authentication (comparison) mode
            is switched, the device should check the compatibility of the card No.*/
            "@opt": ["full", "3Bytes", "4Bytes"]
        },
        "CardVerificationRuleRes": {
            "checkStatus": {
                /*required, string, status of switching card No. authentication (comparison)
                mode: "continue"-switching result can be searched for later, "ok"-switching
                completed, "duplicate"-duplicate data exist and switching failed*/
                "@opt": ["continue", "ok", "duplicate"]
            },
            "progress": {
                /*optional, int, switching progress in percentage which is between 0 and 100,
                and 100 indicates that the card No. authentication (comparison) mode is
                switched*/
                "@min": 0,
                "@max": 100
            }
        }
    }
}
```

## F.73 JSON\_CardVerificationRuleRes

JSON message about the switching progress and configuration result of card No. authentication mode

```
{
    "CardVerificationRuleRes": {
        "checkStatus": "continue",
        /*required, string, status of switching card No. authentication (comparison)
        mode: "continue"-switching result can be searched for later, "ok"-switching
        succeeded, "duplicate"-duplicate data exist and switching failed*/
        "progress": 0
        /*optional, int, switching progress in percentage which is between 0 and 100,
        and 100 indicates that card No. authentication (comparison) mode is switched*/
    }
}
```

## F.74 JSON\_ChannelControllerTypeCfg

JSON message about the device type parameters of the lane controller

```
{
  "ChannelControllerTypeCfg": {
    "deviceModel": ""
    /*required, string, device type: "K3Y501-A"-DS-K3Y501 series flap barrier,
    "K3B501S-A"-DS-K3B501S series swing barrier, "K3B601S-A"-DS-K3B601S series
    swing barrier, "K3G501"-DS-K3G501 series tripod turnstile*/
  }
}
```

## F.75 JSON\_ChannelControllerTypeCfgCap

JSON message about the configuration capability of the lane controller's device type

```
{
  "ChannelControllerTypeCfgCap": {
    "deviceModel": {
      /*required, device type: "K3Y501-A"-DS-K3Y501 series flap barrier, "K3B501S-A"-DS-K3B501S series swing barrier, "K3B601S-A"-DS-K3B601S series swing barrier,
      "K3G501"-DS-K3G501 series tripod turnstile*/
      "@opt": ["K3Y501-A", "K3B501S-A", "K3B601S-A", "K3G501"]
    }
  }
}
```

## F.76 JSON\_ClearAntiSneak

ClearAntiSneak message in JSON format

```
{
  "ClearAntiSneak": {
    "clearAll": ,
    /*required, boolean, whether to clear all anti-passing back records: "true"-yes, "false"-no. Clearing all anti-passing back records is not supported by
    access control devices version 2.1*/
    "EmployeeNoList" : [],
    /*optional, person ID list, this node is valid when clearAll is "false". For
    access control devices version 2.1, if this node is not configured, failure
    response message will be returned*/
    "employeeNo": ""
    /*optional, string, employee No. (person ID)*/
  }
}
```

## F.77 JSON\_ClearAntiSneakCfg

ClearAntiSneakCfg message in JSON format

```
{  
    "ClearAntiSneakCfg":{  
        "ClearFlags":{  
            "antiSneak":  
                /*required, boolean, whether to clear the anti-passing back parameters*/  
            }  
        }  
    }  
}
```

## F.78 JSON\_ClearAttendancePlan

JSON message about the parameters for clearing the attendance schedule

```
{  
    "ClearAttendancePlan":{  
        "ClearFlags":{  
            "attendanceWeekPlan":true,  
            /*optional, boolean, whether to clear the week attendance schedule*/  
            "attendanceTemplate":true  
            /*optional, boolean, whether to clear the parameters of the attendance schedule  
            template*/  
        }  
    }  
}
```

## F.79 JSON\_ClearData

JSON message about the conditions of deleting data from the card

```
{  
    "ClearData":{  
        "checkAll":true,  
        /*optional, boolean, whether to delete all data*/  
        "checkFingerprint":true,  
        /*optional, boolean, whether to delete fingerprint data. This node is valid  
        when the value of checkAll is false or the node checkAll does not exist*/  
        "fingerprints":[1, 2],  
        /*optional, array of int, address list of storage areas where the fingerprints  
        to be deleted are stored. This node is valid when the node checkFingerprint  
        exists. If this node does not exist, it indicates deleting all fingerprints*/  
        "checkFacePicture":true,  
        /*optional, boolean, whether to delete face data. This node is valid when the  
        value of checkAll is false or the node checkAll does not exist*/  
    }  
}
```

```
    "checkCustom":true
/*optional, boolean, whether to delete custom data. This node is valid when the
value of checkAll is false or the node checkAll does not exist*/
    }
}
```

### F.80 JSON\_ClearDataRes

JSON message about the result parameters of deleting data from the card

```
{
  "ClearDataRes":{
    "status":"ok",
    /*required, string, card issuing status: "ok"-succeeded, "failed"-operation
failed, "timeout"-timed out, "verifiyFailure"-authentication failed, "noCard"-no
card detected, "processing"-processing*/
    "cardErrorCode":
    /*dependent, int, internal error code of card operation*/
  }
}
```

### F.81 JSON\_ClearPictureCfg

JSON message about the parameters of clearing all pictures in the device

```
{
  "ClearPictureCfg":{
    "ClearFlags":{
      "facePicture":true,
      /*optional, boolean, whether it supports clearing registered face pictures in
the device*/
      "capOrVerifyPicture":true
      /*optional, boolean, whether it supports clearing authenticated or captured
face pictures in the device*/
    }
  }
}
```

### F.82 JSON\_ClearPictureCfgCap

JSON message about the capability of clearing all pictures in the device

```
{
  "ClearPictureCfgCap":{
    "ClearFlags":{
      "facePicture":{
        /*optional, boolean, whether it supports clearing registered face pictures*/
      }
    }
  }
}
```

```
        "@opt": [true, false]
    },
    "capOrVerifyPicture": {
        /*optional, boolean, whether it supports clearing authenticated or captured
        face pictures*/
        "@opt": [true, false]
    }
}
}
```

## F.83 JSON\_ClearPlansCfg

## ClearPlansCfg message in JSON format

```
{  
    "ClearPlansCfg": {  
        "ClearFlags": {  
            "doorStatusWeekPlan": ,  
/*optional, boolean, whether to clear the week schedule of the door control:  
"true"-yes, "false"-no*/  
            "cardReaderWeekPlan": ,  
/*optional, boolean, whether to clear the week schedule of the card reader  
authentication mode control: "true"-yes, "false"-no*/  
            "userRightWeekPlan": ,  
/*optional, boolean, whether to clear the week schedule of the access  
permission control: "true"-yes, "false"-no*/  
            "doorStatusHolidayPlan": ,  
/*optional, boolean, whether to clear the holiday schedule of the door control:  
"true"-yes, "false"-no*/  
            "cardReaderHolidayPlan": ,  
/*optional, boolean, whether to clear the holiday schedule of the card reader  
authentication mode control: "true"-yes, "false"-no*/  
            "userRightHolidayPlan": ,  
/*optional, boolean, whether to clear the holiday schedule of the access  
permission control: "true"-yes, "false"-no*/  
            "doorStatusHolidayGroup": ,  
/*optional, boolean, whether to clear the holiday group of the door control:  
"true"-yes, "false"-no*/  
            "cardReaderHolidayGroup": ,  
/*optional, boolean, whether to clear the holiday group of the card reader  
authentication mode control: "true"-yes, "false"-no*/  
            "userRightHolidayGroup": ,  
/*optional, boolean, whether to clear the holiday group of the access  
permission control: "true"-yes, "false"-no*/  
            "doorStatusTemplate": ,  
/*optional, boolean, whether to clear the schedule template of the door  
control: "true"-yes, "false"-no*/  
            "cardReaderTemplate": ,  
/*optional, boolean, whether to clear the control schedule template of card
```

```
reader authentication mode: "true"-yes, "false"-no*/
    "userRightTemplate":
/*optional, boolean, whether to clear the schedule template of access
permission control: "true"-yes, "false"-no*/
}
}
}
```

### F.84 JSON\_ControlBlock

JSON message about the control block parameters of a specific section.

```
{
  "ControlBlock": {
    "sectionNo": ,
/*required, integer, section No.*/
    "KeyA": "",
/*optional, string type, a hexadecimal key A password*/
    "KeyB": "",
/*optional, string type, a hexadecimal key B password*/
    "controlBits":""
/*optional, string type, a hexadecimal control bit*/
  }
}
```

### F.85 JSON\_CreateFPLibCond

Message about the conditions of creating face picture library, and it is in JSON format.

```
{
  "faceLibType": "",
/*required, string type, face picture library type: "infraredFD"-infrared face
picture library, "blackFD"-list library, "staticFD"-static library, the maximum
size is 32 bytes*/
  "name": "",
/*required, string type, face picture library name, it cannot be duplicated,
the maximum size is 48 bytes*/
  "customInfo": "",
/*optional, string type, custom information, it is used to indicate the data
property or uniqueness, the maximum size is 192 bytes*/
  "libArmingType": "armingLib",
/*optional, string, library arming type: "armingLib" (arming library),
"nonArmingLib" (non-arming library). Old devices do not support this field. The
default value is "armingLib", that is, all newly created libraries will be
armed by default*/
  "libAttribute": "blackList",
/*optional, string, library attribute type: "general" (normal library),
"blackList" (blocklist library), "VIP" (VIP library), "passerby" (pedestrian
library which cannot be deleted)*/
}
```

```
"FDID": "test"  
/*optional, string, face picture library ID whose maximum string size is 63  
bytes. If this field exists, it indicates that the face picture library ID is  
specified by the platform when the face picture library is created on the  
platform. If this field is not applied by the platform, the face picture  
library ID will be returned by the device. The face picture library ID of the  
same library type is unique*/  
/*Currently the value of FDID can only consist of digits and the number  
converted from this value should be smaller than 2^32*/  
}
```

### F.86 JSON\_CreateFPLibResult

Message about the results of creating face picture library, and it is in JSON format.

```
{  
    "requestURL": "",  
    "statusCode": "",  
    "statusString": "",  
    "subStatusCode": "",  
    "errorCode": "",  
    "errorMsg": "",  
    /*see the description of this node and above nodes in the message of  
    JSON_ResponseStatus*/  
    "FDID": ""  
    /*optional, string type, returned face picture library ID when it created, the  
    library ID of the same type is unique, the maximum length is 63 bytes. This  
    node is valid when errorCode is 1 and errorMsg is "ok"*/  
}
```

#### See Also

[JSON\\_ResponseStatus](#)

### F.87 JSON\_CustomData

JSON message about the conditions of setting custom card information

```
{  
    "CustomData":{  
        "address":1,  
        /*optional, int, start address for writing. By default the data will be written  
        from the start address*/  
        "length":1,  
        /*optional, int, length of the source data to be written, it is 0 by default,  
        unit: byte*/  
        "data":""  
        /*required, string, custom information encoded by Base64*/  
    }  
}
```

```

    }
}
```

## F.88 JSON\_CustomDataRes

JSON message about the result parameters of setting custom card information

```
{
  "CustomDataRes": {
    "status": "ok",
    /*required, string, card issuing status: "ok"-succeeded, "failed"-operation failed, "timeout"-timed out, "verifiyFailure"-authentication failed, "noCard"-no card detected, "processing"-processing*/
    "cardErrorCode": /*dependent, int, internal error code of card operation*/
  }
}
```

## F.89 JSON\_CustomDataResult

JSON message about the results of searching for custom card information

```
{
  "CustomDataResult": {
    "status": "ok",
    /*required, string, card issuing status: "ok"-succeeded, "failed"-operation failed, "timeout"-timed out, "verifiyFailure"-authentication failed, "noCard"-no card detected, "processing"-processing*/
    "cardErrorCode": 0,
    /*dependent, int, internal error code of card operation. This node is valid when the value of status is "failed"*/
    "length": 1,
    /*dependent, int, length of the source data that has been read, unit: byte. This node is valid when the value of status is "ok"*/
    "data": ""
    /*dependent, string, card information encoded by Base64. This node is valid when the value of status is "ok"*/
  }
}
```

## F.90 JSON\_CustomDataSearchCond

JSON message about condition parameters of searching for custom card information

```
{
  "CustomDataSearchCond": {
    "address": 1,
```

```
/*optional, int, start address for reading data. By default the data will be  
read from the start address*/  
    "length":1  
/*optional, int, length of the data that can be read, it is 0 by default, unit:  
byte*/  
    }  
}
```

### F.91 JSON\_DataBlock

JSON message about data block details

```
{  
    "DataBlock": {  
        "addressOfBlock": ,  
/*optional, integer, block address*/  
        "data": "",  
/*required, string, a hexBinary character string, i.e.,  
"f2345678abf2345678abf2345678abf2"*/  
        }  
}
```

### F.92 JSON\_DataBlockCtrl

JSON message about operation parameters of data block

```
{  
    "DataBlockCtrl": {  
        "addressOfBlock": ,  
/*required, integer, block address*/  
        "command": "",  
/*required, string, control commands: "add, minus, copy, paste"*/  
        "value": ,  
/*depend, integer, relative value to be changed, this field is value only when  
the command is set to "add" or "minus"*/  
        }  
}
```

### F.93 JSON\_DataOutputCfg

DataOutputCfg message in JSON format

```
{  
    "DataOutputCfg":{  
        "password": "",  
/*required, string, password for exporting*/  
        "type": ""
```

```
/*optional, string, exporting type: "UsbDisk"-exporting via USB flash drive,  
"UsbPrivate"-exporting via private USB, "ISAPI"-exporting via ISAPI*/  
}  
}
```

### F.94 JSON\_DataOutputProgress

DataOutputProgress message in JSON format

```
{  
    "DataOutputProgress": {  
        "progress":  
/*required, integer, exporting progress*/  
    }  
}
```

### F.95 JSON\_DataTrans

JSON message about data package to be passed through

```
{  
    "DataTrans": {  
        "content": ""  
/*required, string, data to be passed through, which is encoded by Base64*/  
    }  
}
```

### F.96 JSON\_DelFaceRecord

JSON message about the parameters of deleting face records

```
{  
    "FPID": [{  
/*array, list of face record ID, it is the same as the employee No. (person  
ID). Deleting multiple face records in a batch is supported*/  
        "value": ""  
/*required, string type, face record ID, the maximum length is 63 bytes*/  
    }]  
    "operateType": "byTerminal",  
/*optional, string, operation type: "byTerminal"-by terminal*/  
    "terminalNoList": [1]  
/*optional, array, terminal ID list, this node is required when operation type  
is "byTerminal"; currently, only one terminal is supported*/  
}
```

### F.97 JSON\_EditFPlibInfo

Message about the editing information of face picture library, and it is in JSON format.

```
{  
    "name": "",  
    /*optional, face picture library name, string type, the max. string length is  
    48 bytes*/  
    "customInfo": "",  
    /*optional, custom information, it is used to indicate the data property or  
    uniqueness, string type, the max. string length is 192 bytes*/  
    "libArmingType": "armingLib",  
    /*optional, string, arming type of the list library: "armingLib" (armed face  
    picture library), "nonArmingLib" (not armed face picture library). The default  
    value is "armingLib"*/  
    "libAttribute": "blackList"  
    /*optional, string, library type: "blackList" (blocklist library), "VIP" (VIP  
    library), "passerby" (passerby library). The passerby library cannot be  
    deleted*/  
}
```

### F.98 JSON\_EventNotificationAlert\_AccessControllerEvent

The access control event information is uploaded in JSON format of EventNotificationAlert message.

#### Access Control Event Message With Binary Picture Data

```
Content-Type:multipart/form-data;boundary=MIME_boundary  
--MIME_boundary  
Content-Type: application/json  
Content-Length:480  
{  
    "ipAddress": "",  
    /*required, string, IP address of the alarm device, the maximum size is 32  
    bytes*/  
    "ipv6Address": "",  
    /*optional, string, IPv6 address of the alarm device, the maximum size is 128  
    bytes*/  
    "portNo": ,  
    /*optional, integer32, port No. of the alarm device*/  
    "protocol": "",  
    /*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32  
    bytes*/  
    "macAddress": "",  
    /*optional, string, MAC address, the maximum size is 32 bytes*/  
    "channelID": ,  
    /*optional, integer32, device channel No. that triggered the alarm*/
```

```
"dateTime": "",  
/*required, string, time when the alarm is triggered (UTC time), the maximum  
size is 32 bytes*/  
"activePostCount": ,  
/*required, integer32, number of times that the same alarm has been uploaded*/  
"eventType": "",  
/*required, string, triggered event type, here it should be set to  
"AccessControllerEvent", and the maximum size is 128 bytes*/  
"eventState": "",  
/*required, string, event triggering status: "active"-triggered, "inactive"-not  
triggered, the maximum size is 32 bytes*/  
"eventDescription": "",  
/*required, string, event description*/  
"deviceID": "",  
/*optional, string, device No.*/  
"AccessControllerEvent":{  
    "deviceName": "",  
/*optional, string, device name*/  
    "majorEventType": ,  
/*required, int, major alarm/event types (the type value should be converted to  
a decimal number for transmission), see Access Control Event Types for details*/  
    "subEventType": ,  
/*required, int, minor alarm/event types (the type value should be converted to  
a decimal number for transmission), see Access Control Event Types for details*/  
    "inductiveEventType": "",  
/*optional, string, inductive event type. This field is used by storage  
devices; for access control devices, this field is invalid*/  
    "netUser": "",  
/*optional, string, user name for network operations*/  
    "remoteHostAddr": "",  
/*optional, string, remote host address*/  
    "cardNo": "",  
/*optional, string, card No.*/  
    "cardType": ,  
/*optional, int, card type: 1-normal card, 2-disability card, 3-blocklist card,  
4-patrol card, 5-duress card, 6-super card, 7-visitor card, 8-dismiss card*/  
    "name": "",  
/*optional, string, person name*/  
    "whiteListNo": ,  
/*optional, int, allowlist No., which is between 1 and 8*/  
    "reportChannel": ,  
/*optional, int, alarm/event uploading channel type: 1-uploading in arming  
mode, 2-uploading by central group 1, 3-uploading by central group 2*/  
    "cardReaderKind": ,  
/*optional, int, reader type: 1-IC card reader, 2-ID card reader, 3-QR code  
scanner, 4-fingerprint module*/  
    "cardReaderNo": ,  
/*optional, int, reader No.*/  
    "doorNo": ,  
/*optional, int, door or floor No.*/  
    "verifyNo": ,  
/*optional, int, multiple authentication No.*/
```

```

    "alarmInNo": ,
/*optional, int, alarm input No.*/
    "alarmOutNo": ,
/*optional, int, alarm output No.*/
    "caseSensorNo": ,
/*optional, int, event trigger No.*/
    "RS485No": ,
/*optional, int, RS-485 channel No.*/
    "multiCardGroupNo": ,
/*optional, int, group No.*/
    "accessChannel": ,
/*optional, int, turnstile No.*/
    "deviceNo": ,
/*optional, int, device No.*/
    "distractControlNo": ,
/*optional, int, distributed access controller No.*/
    "employeeNo": ,
/*optional, int, employee No. (person ID)*/
    "employeeNoString": "",
/*optional, string, employee No. (person ID). If the field employeeNo exists or
the value of employeeNoString can be converted to that of employeeNo, this
field is required. For the upper-layer platform or client software, the field
employeeNoString will be parsed in prior; if employeeNoString is not
configured, the field employeeNo will be parsed*/
    "localControllerID": ,
/*optional, int, distributed access controller No.: 0-access controller, 1 to
64-distributed access controller No. 1 to distributed access controller No. 64*/
    "InternetAccess": ,
/*optional, int, network interface No.: 1-upstream network interface No. 1, 2-
upstream network interface No. 2, 3-downstream network interface No. 1*/
    "type": ,
/*optional, int, zone type: 0-instant zone, 1-24-hour zone, 2-delayed zone, 3-
internal zone, 4-key zone, 5-fire alarm zone, 6-perimeter zone, 7-24-hour
silent zone, 8-24-hour auxiliary zone, 9-24-hour shock zone, 10-emergency door
open zone, 11-emergency door closed zone, 255-none*/
    "MACAddr": "",
/*optional, string, physical address*/
    "swipeCardType": ,
/*optional, int, card swiping types: 0-invalid, 1-QR code*/
    "serialNo": ,
/*optional, int, event serial No., which is used to check whether the event
loss occurred*/
    "channelControllerID": ,
/*optional, int, lane controller ID: 1-main lane controller, 2-sub-lane
controller*/
    "channelControllerLampID": ,
/*optional, int, light board ID of the lane controller, which is between 1 and
255*/
    "channelControllerIRAdaptorID": ,
/*optional, int, IR adaptor ID of the lane controller, which is between 1 and
255*/
    "channelControllerIREmitterID": ,

```

```
/*optional, int, active infrared intrusion detector No. of the lane controller,  
which is between 1 and 255*/  
    "userType": "",  
/*optional, string, person type: "normal"-normal person (resident), "visitor"-  
visitor, "blacklist"-person in the blocklist, "administrators"-administrator*/  
    "currentVerifyMode": ,  
/*optional, string, current authentication mode of the reader: "cardAndPw"-card  
+password, "card"-card, "cardOrPw"-card or password, "fp"-fingerprint,  
"fpAndPw"-fingerprint+password, "fpOrCard"-fingerprint or card, "fpAndCard"-  
fingerprint+card, "fpAndCardAndPw"-fingerprint+card+password,  
"faceOrFpOrCardOrPw"-face or fingerprint or card or password, "faceAndFp"-face  
+fingerprint, "faceAndPw"-face+password, "faceAndCard"-face+card, "face"-face,  
"employeeNoAndPw"-employee No.+password, "fpOrPw"-fingerprint or password,  
"employeeNoAndFp"-employee No.+fingerprint, "employeeNoAndFpAndPw"-employee No.  
+fingerprint+password, "faceAndFpAndCard"-face+fingerprint+card,  
"faceAndPwAndFp"-face+password+fingerprint, "employeeNoAndFace"-employee No.  
+face, "faceOrfaceAndCard"-face or face+card, "fpOrface"-fingerprint or face,  
"cardOrfaceOrPw"-card or face or password, "cardOrFace"-card or face,  
"cardOrFaceOrFp"-card or face or fingerprint, "cardOrFpOrPw"-card or  
fingerprint or password*/  
    "currentEvent": ,  
/*optional, boolean, whether it is a real-time event: true-yes (real-time  
event), false-no (offline event)*/  
    "QRCodeInfo": "",  
/*optional, string, QR code information*/  
    "thermometryUnit": "",  
/*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheitz"-  
Fahrenheit, "kelvin"-Kelvin*/  
    "currTemperature": ,  
/*optional, float, face temperature which is accurate to one decimal place*/  
    "isAbnormalTemperature": ,  
/*optional, boolean, whether the face temperature is abnormal: true-yes, false-  
no*/  
    "RegionCoordinates": {  
/*optional, face temperature's coordinates*/  
        "positionX": ,  
/*optional, int, normalized X-coordinate which is between 0 and 1000*/  
        "positionY": ,  
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/  
        },  
    "remoteCheck": ,  
/*optional, boolean, whether remote verification is required: true-yes, false-  
no (default)*/  
    "mask": "",  
/*optional, string, whether the person is wearing mask or not: "unknown", "yes"-  
wearing mask, "no"-not wearing mask*/  
    "helmet": "",  
/*optional, string, whether the person is wearing hard hat: "unknown", "yes"-  
wearing hard hat, "no"-not wearing hard hat*/  
    "frontSerialNo": ,  
/*optional, int, the previous event's serial No. If this field does not exist,  
the platform will check whether the event loss occurred according to the field
```

```

serialNo. If both the serialNo and frontSerialNo are returned, the platform will check whether the event loss occurred according to both fields. It is mainly used to solve the problem that the serialNo is inconsistent after subscribing events or alarms*/
    "deviceId": ,
/*optional, string, device's long No., e.g., "10000000101"*/
    "attendanceStatus":"",
/*optional, string, attendance status: "undefined", "checkIn"-check in,
"checkOut"-check out, "breakOut"-break out, "breakIn"-break in, "overtimeIn"-overtime in, "overTimeOut"-overtime out*/
    "statusValue": ,
/*optional, int, status value*/
    "label":"",
/*optional, string, custom attendance name*/
    "pictureURL": "test",
/*optional, string, captured picture URL, size range: [0,256]*/
    "visibleLightURL": "test",
/*optional, string, URL of picture captured by visible light channel, size range: [0,256]*/
    "thermalURL": "test",
/*optional, string, URL of picture captured by thermal imaging channel, size range: [0,256]*/
    "picturesNumber": ,
/*optional, int, number of captured pictures if the capture linkage action is configured. This field will be 0 or not be returned if there is no picture*/
    "purePwdVerifyEnable": ,
/*optional, boolean, whether the device supports opening the door only by password: true-yes, this field is not returned-no. The password used to open the door is the value of the field password in the message
JSON UserInfo
*/
/*For opening the door only by password: 1. The password in "XXX or password" in the authentication mode refers to the person's password (the value of the field password in
JSON UserInfo
); 2. The device will not check the duplication of the password, and the upper platform should ensure that the password is unique; 3. The password cannot be added, deleted, edited, or searched for on the device locally*/
    "HealthInfo": {
/*optional, object, health information*/
        "healthCode": 1,
/*optional, int, health code status: 0 (no request), 1 (no health code), 2 (green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6 (other error, e.g., searching failed due to API exception), 7 (searching for the health code timed out)*/
        "NADCode": 1,
/*optional, int, nucleic acid test result: 0 (no result), 1 (negative, which means normal), 2 (positive, which means diagnosed), 3 (the result has expired)*/
        "travelCode": 1,
/*optional, int, trip code: 0 (no trip in the past 14 days), 1 (once left in the past 14 days), 2 (has been to the high-risk area in the past 14 days), 3

```

```
(other) */
    "vaccineStatus": 1
/*optional, int, whether the person is vaccinated: 0 (not vaccinated), 1
(vaccinated)*/
}
}

--MIME_boundary
Content-Disposition: form-data; name="Picture"; filename="Picture.jpg"; //Captured picture data
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: pictureImage

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data; name="VisibleLight";
filename="VisibleLight.jpg"; //Data of the visible light picture captured by
the thermal camera
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: visibleLight_image

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data; name="Thermal"; filename="Thermal.jpg"; //Thermal picture data
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: thermal_image

fefefwageegfqaeg...
--MIME_boundary
```

### Access Control Event Message With Picture URL

```
{
    "ipAddress": "",
/*required, string, IP address of the alarm device, the maximum size is 32
bytes*/
    "ipv6Address": "",
/*optional, string, IPv6 address of the alarm device, the maximum size is 128
bytes*/
    "portNo": ,
/*optional, integer32, port No. of the alarm device*/
    "protocol": "",
/*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32
bytes*/
    "macAddress": "",
/*optional, string, MAC address, the maximum size is 32 bytes*/
    "channelID": ,
/*optional, integer32, device channel No. that triggered the alarm*/
```

```
"dateTime": "",  
/*required, string, time when the alarm is triggered (UTC time), the maximum  
size is 32 bytes*/  
"activePostCount": ,  
/*required, integer32, number of times that the same alarm has been uploaded*/  
"eventType": "",  
/*required, string, triggered event type, here it should be set to  
"AccessControllerEvent", and the maximum size is 128 bytes*/  
"eventState": "",  
/*required, string, event triggering status: "active"-triggered, "inactive"-not  
triggered, the maximum size is 32 bytes*/  
"eventDescription": "",  
/*required, string, event description*/  
"deviceID": "",  
/*optional, string, device No.*/  
"AccessControllerEvent":{  
    "deviceName": "",  
/*optional, string, device name*/  
    "majorEventType": ,  
/*required, int, major alarm/event types (the type value should be converted to  
a decimal number for transmission), see Access Control Event Types for details*/  
    "subEventType": ,  
/*required, int, minor alarm/event types (the type value should be converted to  
a decimal number for transmission), see Access Control Event Types for details*/  
    "inductiveEventType": "",  
/*optional, string, inductive event type. This field is used by back-end  
devices; for access control devices, this field is invalid*/  
    "netUser": "",  
/*optional, string, user name for network operations*/  
    "remoteHostAddr": "",  
/*optional, string, remote host address*/  
    "cardNo": "",  
/*optional, string, card No.*/  
    "cardType": ,  
/*optional, int, card type: 1-normal card, 2-disability card, 3-blocklist card,  
4-patrol card, 5-duress card, 6-super card, 7-visitor card, 8-dismiss card*/  
    "name": "",  
/*optional, string, person name*/  
    "whiteListNo": ,  
/*optional, int, allowlist No., which is between 1 and 8*/  
    "reportChannel": ,  
/*optional, int, alarm/event uploading channel type: 1-uploading in arming  
mode, 2-uploading by central group 1, 3-uploading by central group 2*/  
    "cardReaderKind": ,  
/*optional, int, reader type: 1-IC card reader, 2-ID card reader, 3-QR code  
scanner, 4-fingerprint module*/  
    "cardReaderNo": ,  
/*optional, int, reader No.*/  
    "doorNo": ,  
/*optional, int, door or floor No.*/  
    "verifyNo": ,  
/*optional, int, multiple authentication No.*/
```

```

    "alarmInNo": ,
/*optional, int, alarm input No.*/
    "alarmOutNo": ,
/*optional, int, alarm output No.*/
    "caseSensorNo": ,
/*optional, int, event trigger No.*/
    "RS485No": ,
/*optional, int, RS-485 channel No.*/
    "multiCardGroupNo": ,
/*optional, int, group No.*/
    "accessChannel": ,
/*optional, int, turnstile No.*/
    "deviceNo": ,
/*optional, int, device No.*/
    "distractControlNo": ,
/*optional, int, distributed access controller No.*/
    "employeeNo": ,
/*optional, int, employee No. (person ID)*/
    "employeeNoString": "",
/*optional, string, employee No. (person ID). If the field employeeNo exists or
the value of employeeNoString can be converted to that of employeeNo, this
field is required. For the upper-layer platform or client software, the field
employeeNoString will be parsed in prior; if employeeNoString is not
configured, the field employeeNo will be parsed*/
    "employeeName": "test",
/*optional, string, employee name. This node is only used for information
release devices, and both this node and the node name should be uploaded*/
    "localControllerID": ,
/*optional, int, distributed access controller No.: 0-access controller, 1 to
64-distributed access controller No. 1 to distributed access controller No. 64*/
    "InternetAccess": "",
/*optional, string, network interface No.: "1"-upstream network interface No.
1, "2"-upstream network interface No. 2, "3"-downstream network interface No.
1*/
    "type": ,
/*optional, int, zone type: 0-instant zone, 1-24-hour zone, 2-delayed zone, 3-
internal zone, 4-key zone, 5-fire alarm zone, 6-perimeter zone, 7-24-hour
client zone, 8-24-hour auxiliary zone, 9-24-hour shock zone, 10-emergency door
open zone, 11-emergency door closed zone, 255-none*/
    "MACAddr": "",
/*optional, string, physical address*/
    "swipeCardType": ,
/*optional, int, card swiping types: 0-invalid, 1-QR code*/
    "serialNo": ,
/*optional, int, event serial No., which is used to check whether the event
loss occurred*/
    "channelControllerID": ,
/*optional, int, lane controller ID: 1-main lane controller, 2-sub lane
controller*/
    "channelControllerLampID": ,
/*optional, int, light board ID of the lane controller, which is between 1 and
255*/

```

```

    "channelControllerIRAdaptorID": ,
/*optional, int, IR adaptor ID of the lane controller, which is between 1 and
255*/
    "channelControllerIREmitterID": ,
/*optional, int, active infrared intrusion detector No. of the lane controller,
which is between 1 and 255*/
    "userType": "",
/*optional, string, person type: "normal"-normal person (resident), "visitor"-visitor,
"blacklist"-person in the blocklist, "administrators"-administrator*/
    "currentVerifyMode": "",
/*optional, string, current authentication mode of the reader: "cardAndPw"-card
+password, "card"-card, "cardOrPw"-card or password, "fp"-fingerprint,
"fpAndPw"-fingerprint+password, "fpOrCard"-fingerprint or card, "fpAndCard"-fingerprint+card,
"fpAndCardAndPw"-fingerprint+card+password,
"faceOrFpOrCardOrPw"-face or fingerprint or card or password, "faceAndFp"-face
+fingerprint, "faceAndPw"-face+password, "faceAndCard"-face+card, "face"-face,
"employeeNoAndPw"-employee No.+password, "fpOrPw"-fingerprint or password,
"employeeNoAndFp"-employee No.+fingerprint, "employeeNoAndFpAndPw"-employee No.
+fingerprint+password, "faceAndFpAndCard"-face+fingerprint+card,
"faceAndPwAndFp"-face+password+fingerprint, "employeeNoAndFace"-employee No.
+face, "faceOrfaceAndCard"-face or face+card, "fpOrface"-fingerprint or face,
"cardOrfaceOrPw"-card or face or password, "cardOrFace"-card or face,
"cardOrFaceOrFp"-card or face or fingerprint, "cardOrFpOrPw"-card or
fingerprint or password*/
    "currentEvent": ,
/*optional, boolean, whether it is a real-time event: true-yes (real-time
event), false-no (offline event)*/
    "QRCodeInfo":"",
/*optional, string, QR code information*/
    "thermometryUnit":"",
/*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheitz"-Fahrenheit,
"kelvin"-Kelvin*/
    "currTemperature": ,
/*optional, float, face temperature which is accurate to one decimal place*/
    "isAbnormalTemperature": ,
/*optional, boolean, whether the face temperature is abnormal: true-yes, false-
no*/
    "RegionCoordinates":{
/*optional, face temperature's coordinates*/
        "positionX": ,
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
        "positionY": ,
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
    },
    "remoteCheck": ,
/*optional, boolean, whether remote verification is required: true-yes, false-
no (default)*/
    "mask":"",
/*optional, string, whether the person is wearing mask or not: "unknown", "yes"-wearing
mask, "no"-not wearing mask*/
    "helmet": "",
/*optional, string, whether the person is wearing hard hat: "unknown", "yes"-*/

```

```

wearing hard hat, "no"-not wearing hard hat*/
    "frontSerialNo": ,
/*optional, int, the previous event's serial No. If this field does not exist,
the platform will check whether the event loss occurred according to the field
serialNo. If both the serialNo and frontSerialNo are returned, the platform
will check whether the event loss occurred according to both fields. It is
mainly used to solve the problem that the serialNo is inconsistent after
subscribing events or alarms*/
    "attendanceStatus":"",
/*optional, string, attendance status: "undefined", "checkIn"-check in,
"checkOut"-check out, "breakOut"-break out, "breakIn"-break in, "overtimeIn"-overtime
in, "overTimeOut"-overtime out*/
    "statusValue": ,
/*optional, int, status value*/
    "label":"",
/*optional, string, custom attendance name*/
    "pictureURL": "",
/*optional, string, captured picture URL*/
    "deviceId": ,
/*optional, string, device's long No., e.g., "10000000101"*/
    "visibleLightURL": "",
/*optional, string, URL of the visible light picture captured by the thermal
camera*/
    "thermalURL": "",
/*optional, string, thermal picture URL*/
    "picturesNumber": ,
/*optional, int, number of captured pictures if the capture linkage action is
configured. This field will be 0 or not be returned if there is no picture*/
    "purePwdVerifyEnable": ,
/*optional, boolean, whether the device supports opening the door only by
password: true-yes, this field is not returned-no. The password used to open
the door is the value of the field password in the message
JSON UserInfo
*/
/*For opening the door only by password: 1. The password in "XXX or password"
in the authentication mode refers to the person's password (the value of the
field password in
JSON UserInfo
);
2. The device will not check the
duplication of the password, and the upper platform should ensure that the
password is unique; 3. The password cannot be added, deleted, edited, or
searched for on the device locally*/
    "HealthInfo": {
/*optional, object, health information*/
        "healthCode": 1,
/*optional, int, health code status: 0 (no request), 1 (no health code), 2
(green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6
(other error, e.g., searching failed due to API exception), 7 (searching for
the health code timed out)*/
        "NADCode": 1,
/*optional, int, nucleic acid test result: 0 (no result), 1 (negative, which
means normal), 2 (positive, which means diagnosed), 3 (the result has expired)*/

```

```
        "travelCode": 1,  
        /*optional, int, trip code: 0 (no trip in the past 14 days), 1 (once left in  
        the past 14 days), 2 (has been to the high-risk area in the past 14 days), 3  
        (other)*/  
        "vaccineStatus": 1  
        /*optional, int, whether the person is vaccinated: 0 (not vaccinated), 1  
        (vaccinated)*/  
    }  
}  
}
```

## See Also

### [Access Control Event Types](#)

#### Example

##### Interaction Example of Uploading Access Control Event with Pictures in Arming Mode

```
HTTP/1.1 200 OK  
MIME-Version: 1.0  
Connection: close  
Content-Type:multipart/form-data;boundary=MIME_boundary  
  
--MIME_boundary  
Content-Type: application/json  
Content-Length:480  
  
<alarm message in JSON format>  
--MIME_boundary  
Content-Disposition: form-data; name="Picture"; filename="Picture.jpg"; //  
Captured picture data  
Content-Type:image/jpeg  
Content-Length:516876  
Content-ID: pictureImage  
  
fefefwageegfqaeg...  
--MIME_boundary  
Content-Type: application/json  
Content-Length:480  
  
<next alarm message in JSON format>  
--MIME_boundary  
Content-Disposition: form-data; name="Picture"; filename="Picture.jpg"  
Content-Type:image/jpeg  
Content-Length:516876  
  
fefefwageegfqaeg...  
--MIME_boundary  
Content-Disposition: form-data; name="VisibleLight";  
filename="VisibleLight.jpg"; //Data of the visible light picture captured by  
the thermal camera  
Content-Type: image/jpeg  
Content-Length: 516876
```

```
Content-ID: visibleLight_image

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data; name="Thermal"; filename="Thermal.jpg"; // Thermal picture data
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: thermal_image

fefefwageegfqaeg...
--MIME_boundary
```

## F.99 JSON\_EventNotificationAlert\_Alarm/EventInfo

EventNotificationAlert message with alarm or event information in JSON format.

```
{
    "ipAddress": "", /*required, device IPv4 address , string, the maximum size is 32 bytes*/
    "ipv6Address": "", /*optional, device IPv6 address, string, the maximum size is 128 bytes*/
    "portNo": , /*optional, device port No., integer32*/
    "protocol": "", /*optional, protocol type, "HTTP, HTTPS", string, the maximum size is 32 bytes*/
    "macAddress": "", /*optional, MAC address, string, the maximum size is 32 bytes, e.g.,
01:17:24:45:D9:F4*/
    "channelID": "", /*optional, device channel No., integer32*/
    "dateTime": "", /*optional, string, alarm/event triggered or occurred time based on ISO8601,
the maximum size is 32 bytes, e.g., 2009-11-14T15:27Z*/
    "activePostCount": "", /*required, alarm/event frequency, integer32*/
    "eventType": "", /*required, alarm/event type, "captureResult, faceCapture,...", string, the
maximum size is 128 bytes*/
    "eventState": "", /*required, string, the maximum size is 32 bytes, durative alarm/event status:
"active"-valid, "inactive"-invalid*/
    "eventDescription": "", /*required, event description, string, the maximum size is 128 bytes*/
    "deviceID": "", /*string type, device ID*/
    "uuid": "", /*string type, event UUID, which is used to uniquely identify an event, the
standard UUID format is xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx*/
    ...
}
```

```
/*optional, for different alarm/event types, the nodes are different, see the  
message examples in different applications*/  
}
```

### F.100 JSON\_EventNotificationAlert\_FaceTempScreeningEventMsg

The event information of face temperature screening is uploaded in JSON format of EventNotificationAlert message.

#### Event Message of Face Temperature Screening with Binary Picture Data

```
{  
    "ipAddress": "",  
    /*required, string, IP address of the alarm device, the maximum size is 32  
    bytes*/  
    "ipv6Address": "",  
    /*optional, string, IPv6 address of the alarm device, the maximum size is 128  
    bytes*/  
    "portNo": ,  
    /*optional, integer32, port No. of the alarm device*/  
    "protocol": "",  
    /*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32  
    bytes*/  
    "macAddress": "",  
    /*optional, string, MAC address, the maximum size is 32 bytes*/  
    "channelID": ,  
    /*optional, integer32, device channel No. that triggered the alarm*/  
    "dateTime": "",  
    /*required, string, time when the alarm is triggered (UTC time), the maximum  
    size is 32 bytes*/  
    "activePostCount": ,  
    /*required, integer32, number of times that the same alarm has been uploaded*/  
    "eventType": "",  
    /*required, string, triggered event type, here it should be set to  
    "FaceTemperatureMeasurementEvent", the maximum size is 128 bytes*/  
    "eventState": "",  
    /*required, string, event triggering status: "active"-triggered, "inactive"-not  
    triggered, the maximum size is 32 bytes*/  
    "eventDescription": "",  
    /*required, string, event description*/  
    "FaceTemperatureMeasurementEvent": {  
        "deviceName": "",  
        /*optional, string, device name*/  
        "serialNo": ,  
        /*optional, int, event serial No.*/  
        "currentEvent": ,  
        /*optional, boolean, whether it is a real-time event: true-yes (real-time  
        event), false-no (offline event)*/  
        "thermometryUnit": "",  
        /*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheith-
```

```
Fahrenheit, "kelvin"-Kelvin*/
    "currTemperature": ,
/*optional, float, face temperature which is accurate to one decimal place*/
    "isAbnormalTemperature": ,
/*optional, boolean, whether the face temperature is abnormal: true-yes, false-
no*/
    "RegionCoordinates":{
/*optional, face temperature's coordinates*/
        "positionX": ,
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
        "positionY": ,
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
        },
    "remoteCheck": ,
/*optional, boolean, whether remote verification is required: true-yes, false-
no (default)*/
    "mask":"",
/*optional, string, whether the person is wearing mask or not: "unknown", "yes"-wearing mask, "no"-not wearing mask*/
    "helmet": ""
/*optional, string, whether the person wears a hard hat: "yes", "no",
"unknown"*/
    }
--MIME_boundary
Content-Disposition: form-data; name="VisibleLight";
filename="VisibleLight.jpg"; //Data of the visible light picture captured by
the thermal camera
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: visibleLight_image

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data; name="Thermal"; filename="Thermal.jpg"; //
Thermal picture data
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: thermal_image

fefefwageegfqaeg...
--MIME_boundaryContent-Disposition: form-data; name="Picture";
filename="Picture.jpg"; //Captured picture data
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: picture_image

fefefwageegfqaeg...
--MIME_boundary
```

## Event Message of Face Temperature Screening with Picture URL

```
{  
    "ipAddress": "",  
    /*required, string, IP address of the alarm device, the maximum size is 32 bytes*/  
    "ipv6Address": "",  
    /*optional, string, IPv6 address of the alarm device, the maximum size is 128 bytes*/  
    "portNo": ,  
    /*optional, integer32, port No. of the alarm device*/  
    "protocol": "",  
    /*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32 bytes*/  
    "macAddress": "",  
    /*optional, string, MAC address, the maximum size is 32 bytes*/  
    "channelID": ,  
    /*optional, integer32, device channel No. that triggered the alarm*/  
    "dateTime": "",  
    /*required, string, time when the alarm is triggered (UTC time), the maximum size is 32 bytes*/  
    "activePostCount": ,  
    /*required, integer32, number of times that the same alarm has been uploaded*/  
    "eventType": "",  
    /*required, string, triggered event type, here it should be set to "FaceTemperatureMeasurementEvent", the maximum size is 128 bytes*/  
    "eventState": "",  
    /*required, string, event triggering status: "active"-triggered, "inactive"-not triggered, the maximum size is 32 bytes*/  
    "eventDescription": "",  
    /*required, string, event description*/  
    "deviceID": "test0123",  
    /*optional, string, device ID (PUID), which should be returned when the event message is uploaded via ISUP*/  
    "FaceTemperatureMeasurementEvent": {  
        "deviceName": "",  
        /*optional, string, device name*/  
        "serialNo": ,  
        /*optional, int, event serial No.*/  
        "currentEvent": ,  
        /*optional, boolean, whether it is a real-time event: true-yes (real-time event), false-no (offline event)*/  
        "thermometryUnit": "",  
        /*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheit"-Fahrenheit, "kelvin"-Kelvin*/  
        "currTemperature": ,  
        /*optional, float, face temperature which is accurate to one decimal place*/  
        "isAbnormalTemperature": ,  
        /*optional, boolean, whether the face temperature is abnormal: true-yes, false-no*/  
        "RegionCoordinates": {
```

```

/*optional, face temperature's coordinates*/
    "positionX": ,
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
    "positionY": ,
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
    },
    "remoteCheck": ,
/*optional, boolean, whether remote verification is required: true-yes, false-no (default)*/
    "mask":"",
/*optional, string, whether the person is wearing mask or not: "unknown", "yes"-wearing mask, "no"-not wearing mask*/
    "visibleLightURL": "",
/*optional, string, URL of the visible light picture captured by the thermal camera*/
    "thermalURL": "",
/*optional, string, thermal picture URL*/
    "pictureURL": "",
/*optional, string, captured picture URL*/
    "helmet": ""
/*optional, string, whether the person wears a hard hat: "yes", "no",
"unknown"*/
}
}

```

### **F.101 JSON\_EventNotificationAlert\_QRCodeEventMsg**

The event information of scanning QR code is uploaded in JSON format of EventNotificationAlert message.

#### **Event Message of Scanning QR Code with Binary Picture Data**

After registering personal information by scanning a fixed QR code using mobile APP, the registered information will be sent to the central storage of the platform, and then the person information URL will be returned which will be used to generate a dynamic QR code in the mobile APP. When a person scans the dynamic QR code on the access control device, the QR code and temperature information will be sent to the platform.

```

{
    "ipAddress":"",
/*required, string, IP address of the alarm device, the maximum size is 32 bytes*/
    "ipv6Address":"",
/*optional, string, IPv6 address of the alarm device, the maximum size is 128 bytes*/
    "portNo": ,
/*optional, integer32, port No. of the alarm device*/
    "protocol":"",
/*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32 bytes*/
}

```

```

    "macAddress":"",
/*optional, string, MAC address, the maximum size is 32 bytes*/
    "channelID": ,
/*optional, integer32, device channel No. that triggered the alarm*/
    "dateTime":"",
/*required, string, time when the alarm is triggered (UTC time), the maximum
size is 32 bytes*/
    "activePostCount": ,
/*required, integer32, number of times that the same alarm has been uploaded*/
    "eventType":"",
/*required, string, triggered event type, here it should be set to
"QRCodeEvent", the maximum size is 128 bytes*/
    "eventState":"",
/*required, string, event triggering status: "active"-triggered, "inactive"-not
triggered, the maximum size is 32 bytes*/
    "eventDescription":"",
/*required, event description*/
    "deviceID": "test0123",
/*optional, string, device ID (PUID), which should be returned when the event
message is uploaded via ISUP*/
    "QRCodeEvent":{
        "deviceName":"",
/*optional, string, device name*/
        "serialNo": ,
/*optional, int, event serial No.*/
        "currentEvent": ,
/*optional, boolean, whether it is a real-time event: true-yes (real-time
event), false-no (offline event)*/
        "QRCodeInfo":"",
/*required, string, QR code information*/
        "thermometryUnit":"",
/*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheitz-
Fahrenheit, "kelvin"-Kelvin*/
        "currTemperature": ,
/*optional, float, face temperature which is accurate to one decimal place*/
        "isAbnormalTemperature": ,
/*optional, boolean, whether the face temperature is abnormal: true-yes, false-
no*/
        "RegionCoordinates":{
/*optional, face temperature's coordinates*/
            "positionX": ,
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
            "positionY": ,
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
        },
        "remoteCheck": ,
/*optional, boolean, whether remote verification is required: true-yes, false-
no (default)*/
        "mask":"",
/*optional, string, whether the person is wearing mask or not: "unknown", "yes"-wearing
mask, "no"-not wearing mask*/
        "helmet": ""
    }

```

```
/*optional, string, whether the person wears a hard hat: "yes", "no",
"unknown"*/
}
--MIME_boundary
Content-Disposition: form-data; name="VisibleLight";
filename="VisibleLight.jpg"; //Data of the visible light picture captured by
the thermal camera
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: visibleLight_image

fefefwageegfqaeg...
--MIME_boundary
Content-Disposition: form-data; name="Thermal"; filename="Thermal.jpg"; //
Thermal picture data
Content-Type: image/jpeg
Content-Length: 516876
Content-ID: thermal_image

fefefwageegfqaeg...
--MIME_boundary
```

### Event Message of Scanning QR Code with Picture URL

```
{
    "ipAddress": "",  

    /*required, string, IP address of the alarm device, the maximum size is 32  

bytes*/
    "ipv6Address": "",  

    /*optional, string, IPv6 address of the alarm device, the maximum size is 128  

bytes*/
    "portNo": ,  

    /*optional, integer32, port No. of the alarm device*/
    "protocol": "",  

    /*optional, string, protocol type: "HTTP", "HTTPS", the maximum size is 32  

bytes*/
    "macAddress": "",  

    /*optional, string, MAC address, the maximum size is 32 bytes*/
    "channelID": ,  

    /*optional, integer32, device channel No. that triggered the alarm*/
    "dateTime": "",  

    /*required, string, time when the alarm is triggered (UTC time), the maximum  

size is 32 bytes*/
    "activePostCount": ,  

    /*required, integer32, number of times that the same alarm has been uploaded*/
    "eventType": "",  

    /*required, string, triggered event type, here it should be set to  

"QRCodeEvent", the maximum size is 128 bytes*/
    "eventState": "",  

    /*required, string, event triggering status: "active"-triggered, "inactive"-not  

triggered, the maximum size is 32 bytes*/
```

```

    "eventDescription":"",
/*required, event description*/
    "deviceID": "test0123",
/*optional, string, device ID (PUID), which should be returned when the event
message is uploaded via ISUP*/
    "QRCodeEvent":{
        "deviceName":"",
/*optional, string, device name*/
        "serialNo": ,
/*optional, int, event serial No.*/
        "currentEvent": ,
/*optional, boolean, whether it is a real-time event: true-yes (real-time
event), false-no (offline event)*/
        "QRCodeInfo":"",
/*required, string, QR code information*/
        "thermometryUnit":"",
/*optional, string, temperature unit: "celsius"-Celsius (default), "fahrenheit"-Fahrenheit, "kelvin"-Kelvin*/
        "currTemperature": ,
/*optional, float, face temperature which is accurate to one decimal place*/
        "isAbnormalTemperature": ,
/*optional, boolean, whether the face temperature is abnormal: true-yes, false-
no*/
        "RegionCoordinates":{
/*optional, face temperature's coordinates*/
            "positionX": ,
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
            "positionY": ,
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
        },
        "remoteCheck": ,
/*optional, boolean, whether remote verification is required: true-yes, false-
no (default)*/
        "mask":"",
/*optional, string, whether the person is wearing mask or not: "unknown", "yes"-wearing mask, "no"-not wearing mask*/
        "visibleLightURL": "",
/*optional, string, URL of the visible light picture captured by the thermal
camera*/
        "thermalURL": "",
/*optional, string, thermal picture URL*/
        "helmet": ""
/*optional, string, whether the person wears a hard hat: "yes", "no",
"unknown"*/
    }
}

```

### **F.102 JSON\_EventOptimizationCfg**

EventOptimizationCfg message in JSON format

```
{
    "EventOptimizationCfg": {
        "enable": ,
        /*optional, boolean, whether to enable event optimization: true-yes (default),
        false-no*/
        "isCombinedLinkageEvents":
        /*optional, boolean, whether to enable linked event combination: true-enable
        (default), false-disable*/
    }
}
```

### F.103 JSON\_EventStorageCfg

JSON message about the storage parameters of access control events

```
{
    "EventStorageCfg": {
        "mode": "regular",
        /*required, string, event storage method: "regular" (delete old events
        periodically), "time" (delete old events by specified time), "cycle"
        (overwriting)*/
        "checkTime": "",
        /*dependent, string, check time. Events that occurred before the check time
        will be deleted. The maximum size is 32 bytes. This node is valid when mode is
        "time"*/
        "period": 10
        /*dependent, int, time period for deleting old events, unit: minute. This node
        is valid when mode is "regular"*/
    }
}
```

### F.104 JSON\_EventStorageCfgCap

JSON message about the storage configuration capability of access control events

```
{
    "EventStorageCfgCap": {
        "mode": {
            /*required, string, event storage method: "regular" (delete old events
            periodically), "time" (delete old events by specified time), "cycle"
            (overwriting)*/
            "@opt": ["regular", "time", "cycle"]
        },
        "checkTime": {
            /*dependent, string, check time. Events that occurred before the check time
            will be deleted. The maximum size is 32 bytes. This node is valid when mode is
            "time"*/
            "@min": 0,
        }
    }
}
```

```

        "@max":0
    },
    "period":{
/*dependent, int, time period for deleting old events, unit: minute. This node
is valid when mode is "regular"*/
        "@min":10,
        "@max":10
    }
}
}

```

### **F.105 JSON\_FaceRecognizeMode**

FaceRecognizeMode message in JSON format

```

{
    "FaceRecognizeMode":{
/*required, facial recognition mode: "normalMode"-normal mode, "deepMode"-deep
mode*/
        "mode":"""
    }
}

```

### **F.106 JSON\_FaceRecordNumInAllFPLib**

Message about the total number of face records in all face picture libraries, and it is in JSON format.

```

{
    "requestURL": "",
    "statusCode": "",
    "statusString": "",
    "subStatusCode": "",
    "errorCode": "",
    "errorMsg": "",
/*see the description of this node and above nodes in the message of
JSON_ResponseStatus*/
    "FDRecordDataInfo": [
/*optional, string type, information of face records in face picture library,
this node is valid when errorCode is 1 and errorMsg is "ok"*/
        "FDID": "",
/*optional, face picture library ID, string type, the maximum size is 63 bytes*/
        "faceLibType": "",
/*optional, face picture library type: "blackFD"-list library, "staticFD"-static
library, string type, the maximum size is 32 bytes*/
        "name": "",
/*optional, face picture library name, string type, the maximum size is 48
bytes*/
}

```

```
    "recordDataNumber": ""  
/*optional, number of records, integer32 type*/  
    }]  
}
```

### See Also

[JSONResponseStatus](#)

## F.107 JSON\_FaceRecordNumInOneFPLib

Message about the number of face records in a specific face picture library, and it is in JSON format.

```
{  
    "requestURL": "",  
    "statusCode": "",  
    "statusString": "",  
    "subStatusCode": "",  
    "errorCode": "",  
    "errorMsg": "",  
/*see the description of this node and above nodes in the message of  
JSON_ResponseStatus*/  
    "FDID": "",  
/*optional, face picture library ID, string type, the max. string length is 63  
bytes*/  
    "faceLibType": "",  
/*optional, face picture library type: "blackFD"-list library, "staticFD"-  
static library, string type, the max. string length is 32 bytes*/  
    "name": "",  
/*optional, face picture library name, string type, the max. string length is  
48 bytes*/  
    "recordDataNumber": ""  
/*optional, number of records, integer32 type*/  
}
```

### See Also

[JSONResponseStatus](#)

## F.108 JSON\_FaceTemperatureEvent

JSON message about the result of actively getting face temperature screening events

```
{  
    "FaceTemperatureEvent" : {  
        "searchID": "",  
/*required, string, search ID, which is used to check whether the current  
search requester is the same as the previous one. If they are the same, the
```

```

search record will be stored in the device to speed up the next search*/
    "responseStatusStrg": "OK",
/*required, string, search status: "OK"(searching completed), "MORE"(searching
for more results), "NO MATCH"(no matched results)*/
    "numOfMatches": 1,
/*required, int, the number of the returned records*/
    "totalMatches": 1,
/*required, int, the total number of the matched records*/
    "InfoList": [
/*optional, event information*/
        "deviceName": "",
/*optional, string, device name*/
        "serialNo": 1,
/*optional, int, event serial No.*/
        "thermometryUnit": "",
/*required, string, temperature unit: "celsius" (Celsius), "fahrenheit"
(Fahrenheit), "kelvin" (Kelvin), the default value is "celsius"*/
        "currTemperature": 1.0,
/*required, float, face temperature, the value is accurate to one decimal
place*/
        "isAbnormalTemperature": true,
/*optional, boolean, whether the face temperature is abnormal: true (abnormal),
false (normal)*/
        "RegionCoordinates": {
/*optional, coordinates of the face temperature*/
            "positionX": 1,
/*optional, int, X-coordinate, which is normalized to a number between 0 and
1000*/
            "positionY": 1
/*optional, int, Y-coordinate, which is normalized to a number between 0 and
1000*/
        },
        "mask": ""
/*optional, string, whether the person wears a mask: "unknown" (unknown), "yes"
(wearing a mask), "no" (no mask)*/
        "capturePicUrl":"",
/*optional, string, the URL of the captured picture*/
        "visibleLightPicUrl":"",
/*optional, string, the URL of the visible light picture*/
        "thermalPicUrl":"",
/*optional, string, the URL of the thermal picture*/
        "helmet": "",
/*optional, string, whether the person wears a hard hat: "unknown" (unknown),
"yes" (wearing a hard hat), "no" (no hard hat)*/
        "dateTime": "2016-12-12T17:30:08+08:00"
/*required, string, the time (UTC time) when the alarm is triggered, the
maximum size is 32 bytes*/
    }
}
}

```

## F.109 JSON\_FaceTemperatureEventCap

JSON message about the capability of actively getting face temperature screening events

```
{
    "FaceTemperatureEventCap" : {
        "FaceTemperatureEventCond": {
            "searchID": {
                /*required, string, search ID, which is used to check whether the current
                search requester is the same as the previous one. If they are the same, the
                search record will be stored in the device to speed up the next search*/
                "@min":1,
                "@max":1
            },
            "searchResultPosition": {
                /*required, int, the start position of search result in the result list. In a
                single search, if you cannot get all the records in the result list, you can
                mark the end position and get the following records after the marked position
                in the next search. If the maximum number of totalMatches supported by the
                device is M and the number of totalMatches stored in the device now is N
                (N<=M), the valid range of this node is 0 to N-1*/
                "@min":1,
                "@max":1
            },
            "maxResults": {
                /*required, int, the maximum number of search results that can be obtained by
                calling the URI this time. If the value of maxResults is greater than that
                defined in the device capability, the value in the capability will be returned.
                In this case, the device will not return error*/
                "@min":1,
                "@max":1
            },
            "startTime": "1970-01-01T00:00:00+00:00",
            /*optional, string, start time (UTC time)*/
            "endTime": "2017-12-12T17:30:08+08:00",
            /*optional, string, end time (UTC time)*/
            "picEnable": {
                /*optional, boolean, whether to upload the picture along with the event
                information: true (all matched events will be uploaded with pictures if there
                are any), false (all matched events will be uploaded without pictures). If this
                node is not configured, the default value is true*/
                "@opt": [true, false]
            },
            "beginSerialNo": {
                /*optional, int, start serial No.*/
                "@min":1,
                "@max":1
            },
            "endSerialNo": {
                /*optional, int, end serial No.*/
                "@min":1,
                "@max":1
            }
        }
    }
}
```

```

        "@max":1
    },
    "isAbnormalTemperature":{
/*optional, object, whether the skin-surface temperature is abnormal*/
        "@opt":[true, false]
/*optional, array of boolean, options: true (yes), false (no)*/
    }
},
"InfoList" : {
/*optional, event information*/
    "deviceName": {
/*optional, string, device name*/
        "@min":1,
        "@max":1
    },
    "serialNo": {
/*optional, int, event serial No.*/
        "@min":1,
        "@max":1
    },
    "thermometryUnit": {
/*required, string, temperature unit: "celsius" (Celsius), "fahrenheit"
(Fahrenheit), "kelvin" (Kelvin), the default value is "celsius"*/
        "@opt":["celsius","fahrenheit","kelvin"]
    },
    "currTemperature": {
/*required, float, face temperature, the value is accurate to one decimal
place*/
        "@min":1.0,
        "@max":1.0
    },
    "isAbnormalTemperature": {
/*optional, boolean, whether the face temperature is abnormal: true (abnormal),
false (normal)*/
        "@opt":[true, false]
    },
    "RegionCoordinates": {
/*optional, coordinates of the face temperature*/
        "positionX": {
/*optional, int, X-coordinate, which is normalized to a number between 0 and
1000*/
            "@min":1,
            "@max":1
        },
        "positionY": {
/*optional, int, Y-coordinate, which is normalized to a number between 0 and
1000*/
            "@min":1,
            "@max":1
        }
    },
    "mask": {

```

```

/*optional, string, whether the person wears a mask: "unknown" (unknown), "yes"
(wearing a mask), "no" (no mask)*/
    "@opt": ["unknown", "yes", "no"]
},
"capturePicUrl": {
/*optional, string, the URL of the captured picture*/
    "@min": 1,
    "@max": 1
},
"visibleLightPicUrl": {
/*optional, string, the URL of the visible light picture*/
    "@min": 1,
    "@max": 1
},
"thermalPicUrl": {
/*optional, string, the URL of the thermal picture*/
    "@min": 1,
    "@max": 1
},
"helmet": {
/*optional, string, whether the person wears a hard hat: "unknown" (unknown),
"yes" (wearing a hard hat), "no" (no hard hat)*/
    "@opt": ["unknown", "yes", "no"]
},
"dateTime": "2016-12-12T17:30:08+08:00"
/*required, string, the time (UTC time) when the alarm is triggered, the
maximum size is 32 bytes*/
}
}
}

```

### F.110 JSON\_FaceTemperatureEventCond

JSON message about the condition of actively getting face temperature screening events

```

{
  "FaceTemperatureEventCond": {
    "searchID": "",
/*required, string, search ID, which is used to check whether the current
search requester is the same as the previous one. If they are the same, the
search record will be stored in the device to speed up the next search*/
    "searchResultPosition": 0,
/*required, int, the start position of search result in the result list. In a
single search, if you cannot get all the records in the result list, you can
mark the end position and get the following records after the marked position
in the next search. If the maximum number of totalMatches supported by the
device is M and the number of totalMatches stored in the device now is N
(N<=M), the valid range of this node is 0 to N-1*/
    "maxResults": 30,
/*required, int, the maximum number of search results that can be obtained by

```

```

calling the URI this time. If the value of maxResults is greater than that
defined in the device capability, the value in the capability will be returned.
In this case, the device will not return error*/
    "startTime": "2016-12-12T17:30:08+08:00",
/*optional, string, start time (UTC time)*/
    "endTime": "2017-12-12T17:30:08+08:00",
/*optional, string, end time (UTC time)*/
    "picEnable": true,
/*optional, boolean, whether to upload the picture along with the event
information: true (all matched events will be uploaded with pictures if there
are any), false (all matched events will be uploaded without pictures). If this
node is not configured, the default value is true*/
    "beginSerialNo": 1,
/*optional, int, start serial No.*/
    "endSerialNo": 1,
/*optional, int, end serial No.*/
    "isAbnormalTemperature": true
/*optional, boolean, whether the skin-surface temperature is abnormal*/
}
}

```

### F.111 JSON\_FingerPrintCfg

FingerPrintCfg message in JSON format

```

{
  "FingerPrintCfg": {
    "employeeNo": "",
/*required, string, employee No. (person ID) linked with the fingerprint*/
    "enableCardReader": ,
/*required, array, fingerprint modules to apply fingerprint data to, e.g.,
[1,3,5] indicates applying fingerprint data to fingerprint modules No.1, No.3,
and No.5*/
    "fingerPrintID": ,
/*required, integer, fingerprint No., which is between 1 and 10*/
    "deleteFingerPrint": ,
/*optional, boolean, whether to delete the fingerprint: "true"-yes. This node
is required only when the fingerprint needs to be deleted; for adding or
editing fingerprint information, this node can be set to NULL*/
    "fingerType": "",
/*required, string, fingerprint type: "normalFP"-normal fingerprint, "hijackFP"-  
duress fingerprint, "patrolFP"-patrol fingerprint, "superFP"-super fingerprint,
"dismissingFP"-dismiss fingerprint*/
    "fingerData": "",
/*required, string, fingerprint data encoded by Base64*/
    "leaderFP": ,
/*optional, array, whether the access control points support first fingerprint
authentication function, e.g., [1,3,5] indicates that access control points No.
1, No.3, and No.5 support first fingerprint authentication function*/
    "checkEmployeeNo": 
  }
}

```

```

/*optional, boolean, whether to check the existence of the employee No. (person
ID): "false"-no, "true"-yes. If this node is not configured, the device will
check the existence of the employee No. (person ID) by default. If this node is
set to "false", the device will not check the existence of the employee No.
(person ID) to speed up data applying; if this node is set to "true" or NULL,
the device will check the existence of the employee No. (person ID), and it is
recommended to set this node to "true" or NULL if there is no need to speed up
data applying*/
    }
}

```

### F.112 JSON\_FingerPrintCond

FingerPrintCond message in JSON format

```

{
  "FingerPrintCond": {
    "searchID": "",
    /*required, string, search ID, which is used to confirm the upper-level
    platform or system. If the platform or the system is the same one during two
    searching, the search history will be saved in the memory to speed up next
    searching*/
    "employeeNo": "",
    /*required, string, employee No. (person ID) linked with the fingerprint*/
    "cardReaderNo": ,
    /*optional, integer, fingerprint module No.*/
    "fingerPrintID": ,
    /*optional, integer, fingerprint No., which is between 1 and 10*/
  }
}

```

### F.113 JSON\_FingerPrintDelete

FingerPrintDelete message in JSON format

```

{
  "FingerPrintDelete": {
    "mode": "",
    /*required, string, deleting mode: "byEmployeeNo"-delete by employee No.
    (person ID), "byCardReader"-delete by fingerprint module*/
    "EmployeeNoDetail": {
      /*optional, delete by employee No. (person ID), this node is valid when mode is
      "byEmployeeNo"*/
      "employeeNo": "",
      /*optional, string, employee No. (person ID) linked with the fingerprint*/
      "enableCardReader": ,
      /*optional, array, fingerprint module whose fingerprints should be deleted,
      e.g., [1,3,5] indicates that the fingerprints of fingerprint modules No.1, No.
      3, and No.5 are deleted*/
    }
  }
}

```

```

        "fingerPrintID":  

/*optional, array, No. of fingerprint to be deleted, e.g., [1,3,5] indicates  

deleting fingerprint No.1, No.3, and No.5*/  

    },  

    "CardReaderDetail":{  

/*optional, delete by fingerprint module, this node is valid when mode is  

"byCardReader"*/  

        "cardReaderNo": ,  

/*optional, integer, fingerprint module No.*/  

        "clearAllCard": ,  

/*optional, boolean, whether to delete the fingerprint information of all  

cards: "false"-no (delete by employee No.), "true"-yes (delete the fingerprint  

information of all employee No.)*/  

        "employeeNo":""  

/*optional, string, employee No. (person ID) linked with the fingerprint, this  

node is valid when clearAllCard is "false"*/  

    }  

}  

}

```

### **F.114 JSON\_FingerPrintDeleteProcess**

FingerPrintDeleteProcess message in JSON format

```
{
  "FingerPrintDeleteProcess":{  

    "status":""
/*required, string, deleting status: "processing"-deleting, "success"-deleted,  

"failed"-deleting failed*/
  }
}
```

### **F.115 JSON\_FingerPrintInfo**

FingerPrintInfo message in JSON format

```
{
  "FingerPrintInfo":{  

    "searchID":"",
/*required, string, search ID, which is used to confirm the upper-level  

platform or system. If the platform or the system is the same one during two  

searching, the search history will be saved in the memory to speed up next  

searching*/
    "status":"",
/*required, string, status: "OK"-the fingerprint exists, "NoFP"-the fingerprint  

does not exist*/
    "FingerPrintList":[]  

      "cardReaderNo": ,
/*required, integer, fingerprint module No.*/
  }
}
```

```
        "fingerPrintID": ,
/*required, integer, fingerprint No., which is between 1 and 10*/
        "fingerType":"",
/*required, string, fingerprint type: "normalFP"-normal fingerprint, "hijackFP"-duress fingerprint, "patrolFP"-patrol fingerprint, "superFP"-super fingerprint, "dismissingFP"-dismiss fingerprint*/
        "fingerData":"",
/*required, string, fingerprint data encoded by Base64*/
        "leaderFP":
/*optional, array, whether the access control points support first fingerprint authentication function, e.g., [1,3,5] indicates that access control points No. 1, No.3, and No.5 support first fingerprint authentication function*/
    }]
}
}
```

### F.116 JSON\_FingerPrintModify

FingerPrintModify message in JSON format

```
{
  "FingerPrintModify":{
    "employeeNo":"",
/*required, string, employee No. (person ID) linked with the fingerprint*/
    "cardReaderNo": ,
/*required, integer, fingerprint module No.*/
    "fingerPrintID": ,
/*required, integer, fingerprint No., which is between 1 and 10*/
    "fingerType":"",
/*required, string, fingerprint type: "normalFP"-normal fingerprint, "hijackFP"-duress fingerprint, "patrolFP"-patrol fingerprint, "superFP"-super fingerprint, "dismissingFP"-dismiss fingerprint. If this node is not configured, the fingerprint type will be the original type*/
    "leaderFP": ,
/*optional, array, whether the access control points support first fingerprint authentication function, e.g., [1,3,5] indicates that access control points No. 1, No.3, and No.5 support first fingerprint authentication function. If this node is not configured, the first fingerprint authentication function will remain unchanged*/
  }
}
```

### F.117 JSON\_FingerPrintStatus

FingerPrintStatus message in JSON format

```
{
  "FingerPrintStatus":{
    "status":"",

```

```

/*optional, string, status: "success", "failed". This node will be returned
only when editing fingerprint parameters or deleting fingerprints; for applying
fingerprint data to the fingerprint module, this node will not be returned*/
    "StatusList": [
/*optional, status list. This node will be returned only when applying
fingerprint data to the fingerprint module; for editing fingerprint parameters
or deleting fingerprints, this node will not be returned*/
        "id": ,
/*optional, integer, fingerprint module No.*/
        "cardReaderRecvStatus": ,
/*optional, integer, fingerprint module status: 0-connecting failed, 1-
connected, 2-the fingerprint module is offline, 3-the fingerprint quality is
poor, try again, 4-the memory is full, 5-the fingerprint already exists, 6-the
fingerprint ID already exists, 7-invalid fingerprint ID, 8-this fingerprint
module is already configured, 10-the fingerprint module version is too old to
support the employee No.*/
        "errorMsg": "",
/*optional, string, error information*/
        ],
        "totalStatus": ,
/*required, integer, applying status: 0-applying, 1-applied*/
    }
}

```

### F.118 JSON\_FPLibCap

Face picture library capability message, and it is in JSON format.

```

{
    "requestURL": "",
    "statusCode": ,
    "statusString": "",
    "subStatusCode": "",
    "errorCode": ,
    "errorMsg": " ",
/*see the description of this node and the above nodes in the message of
JSON_ResponseStatus*/
    "FDNameMaxLen": ,
/*required, integer32 type, maximum length of face picture library name, the
default value is 64 bytes*/
    "customInfoMaxLen": ,
/*required, int, maximum length of custom information, the default value is 256
bytes, read-only*/
    "FDMaxNum": ,
/*required, integer32 type, maximum number of face picture libraries, the
default value is 3*/
    "FDRecordDataMaxNum": ,
/*required, integer type, maximum face records supported by face picture
library*/
    "supportFDFunction": "post,delete,put,get,setUp",
}

```

```
/*required, the supported operations on face picture library: "post"-create,
"delete"-delete, "put"-edit, "get"-search, "setUp"-set*/
    "isSupportFDSearch": ,
/*required, boolean type, whether supports searching in face picture library:
"true"-yes, "false"-no*/
    "isSupportFDSearchDataPackage": ,
/*required, boolean type, whether supports packaging the found data in the face
picture library: "true"-yes, "false"-no*/
    "isSupportFSsearchByPic": ,
/*required, boolean type, whether supports searching by picture in the face
picture library: "true"-yes, "false"-no*/
    "isSupportFSsearchByPicGenerate": ,
/*required, boolean type, whether supports exporting search by picture results
from the face picture library: "true"-yes, "false"-no*/
    "isSupportFDSearchDuplicate": ,
/*required, boolean type, whether supports duplication checking: "true"-yes,
"false"-no*/
    "isSupportFDSearchDuplicateGenerate": ,
/*required, boolean type, whether supports exporting the duplication checking
results: "true"-yes, "false"-no*/
    "isSupportFCSearch": ,
/*required, boolean type, whether supports searching face picture comparison
alarms: "true"-yes, "false"-no*/
    "isSupportFCSearchDataPackage": ,
/*required, boolean, whether supports packaging the search results of face
picture comparison alarms: "true"-yes, "false"-no*/
    "isSupportFDExecuteControl": ,
/*required, boolean, whether supports creating relation between face picture
libraries and cameras: "true"-yes, "false"-no*/
    "generateMaxNum": ,
/*required, integer32 type, maximum face records can be exported from face
picture library*/
    "faceLibType":"blackFD,staticFD,infraredFD",
/*optional, string type, face picture library types: "blackFD"-list library,
"staticFD"-static library, "infraredFD"-infrared face picture library, the
maximum size of value can be assigned to this node is 32 bytes*/
    "modelMaxNum": ,
/*optional, integer type, the maximum number of search results, the default
value is 100*/
    "isSupportModelData":true,
/*optional, boolean type, whether to support applying model data: "true"-yes,
this node is not returned-no*/
    "isSupportFDLibArmingType": ,
/*optional, boolean, whether it supports face picture library arming type:
true, false*/
    "isSupportFDLibSearch": ,
/*optional, boolean, whether it supports searching face picture library: true,
false*/
    "FDArmingRecordDataMaxNum": ,
/*optional, integer32, the supported maximum number of face records in the face
picture arming library*/
    "isSupportControlPersonRecordByHumanId": ,
```

```

/*optional, boolean, whether it supports modifying and deleting the face record
by humanId: true, false*/
    "isSupportControlPersonRecordByRowKey": ,
/*optional, boolean, whether it supports modifying and deleting the face record
by rowKey: true, false*/
    "isSupportFaceLibRebuildCfg": ,
/*optional, boolean, whether it supports recreating face picture library
information and configuration: true, false*/
    "isSupportFDMove": ,
/*optional, boolean, whether it supports moving face data in the face picture
library in a batch: true, false. The related URI is /ISAPI/Intelligent/FDLib/
FDMove/capabilities?format=json*/
    "faceURLLen": ,
/*optional, int, the maximum size of the face picture URL. If this node is not
returned, the default size of the face picture URL supported by the device is
256 bytes; otherwise, the device should support that the value of this node is
greater than or equal to 256*/
    "featurePointTypeList":
["face","leftEye","rightEye","leftMouthCorner","rightMouthCorner","nose"],
/*optional, array of string, feature point types of face pictures supported by
the device. If this node exists, it indicates that the device supports applying
feature points of pictures, and the returned values are feature point types
supported by the device*/
    "isSupportArmingLibCfg":true,
/*optional, boolean, whether it supports configuring parameters of the armed
face picture library, read-only, related URI: /ISAPI/Intelligent/FDLib/
armingLibCfg/capabilities?format=json*/
    "isSupportModelTransformation":true,
/*optional, boolean, whether it supports converting face picture models in the
face picture list library, read-only, related URI: /ISAPI/Intelligent/FDLib/
model/transformation/capabilities?format=json*/
    "libAttribute":{
/*optional, object, library type: "general" (general library), "blackList"
(blocklist library), "VIP" (VIP library), "passerby" (strange library which
cannot be deleted)*/
        "@opt":["general", "blackList", "VIP", "passerby"]
/*optional, array of string, options*/
    },
    "faceType":{
/*optional, object, face type*/
        "@opt":["normalFace", "patrolFace", "hijackFace", "superFace"]
    },
    "saveFacePic": {
/*optional, object, whether to save face pictures*/
        "@opt": [true, false]
    },
    "leaderPermission":{
/*optional, object, first authentication permission*/
        "@size":4,
/*optional, int, the maximum number of elements in the array, value range:
[1,4]*/
        "@min":1,

```

```
/*optional, int, the minimum value of the element, value range: [1, 4]*/
    "@min":1
    /*optional, int, the maximum value of the element, value range: [1, 4]*/
    "@max":4
}
```

### See Also

[JSONResponseStatus](#)

## F.119 JSON\_FPLibListInfo

Message about the list of face picture libraries, and it is in JSON format.

```
{
    "requestURL": "",
    "statusCode": "",
    "statusString": "",
    "subStatusCode": "",
    "errorCode": "",
    "errorMsg": "",
    /*see the description of this node and above nodes in the message of
    JSON_ResponseStatus*/
    "FDLib": [
        /*optional, face picture library information, string type, this node is valid
        when errorCode is 1 and errorMsg is "ok"*/
        {
            "FDID": "",
            /*optional, face picture library ID, string type, the maximum size is 63 bytes*/
            "faceLibType": "",
            /*optional, face picture library type: "blackFD"-list library, "staticFD"-static
            library, string type, the maximum size is 32 bytes*/
            "name": "",
            /*optional, face picture library name, string type, the maximum size is 48
            bytes*/
            "customInfo": ""
            /*optional, custom information, string type, the maximum size is 192 bytes*/
        }
    ]
}
```

### See Also

[JSONResponseStatus](#)

## F.120 JSON\_HealthCodeCfg

JSON message about the health code parameters

```
{  
    "enabled":true,  
    /*required, boolean, whether to enable the health code*/  
    "serverAddress":"test"  
    /*optional, string, address of the health code server, the maximum string size  
    is 128 bytes*/  
}
```

### F.121 JSON\_HealthCodeDisplayCfg

JSON message about the health code display parameters

```
{  
    "showHealthCode":true  
    /*required, boolean, whether to display the health code: true, false*/  
}
```

### F.122 JSON\_IDCardInfoEvent

JSON message about the result of getting ID card swiping events actively

```
{  
    "IDCardInfoEvent":{  
        "searchID":"",
        /*required, string, search ID, which is used to check whether the current
        search requester is the same as the previous one. If they are the same, the
        search record will be stored in the device to speed up the next search*/
        "responseStatusStrg":"OK",
        /*required, string, searching status: "OK" (searching completed), "MORE"
        (search for more data), "NO MATCH" (no matched data)*/
        "numOfMatches":1,
        /*required, int, number of records returned this time*/
        "totalMatches":1,
        /*required, int, total number of matched records*/
        "InfoList":[]  
        /*optional, event information*/
        "deviceName":"",
        /*optional, string, device name*/
        "major":1,
        /*required, int, major event type, 0 means all event types. For details, refer
        to Access Control Event Types. The value of this node is in decimal format
        instead of hexadecimal format (for example, 1 refers to 0x1 which indicates
        that the major type is MAJOR_ALARM)*/
        "minor":1,
        /*required, int, minor event type, 0 means all event types. For details, refer
        to Access Control Event Types. The value of this node is in decimal format
        instead of hexadecimal format (for example, 1024 refers to 0x400 which
        indicates that the minor type is MINOR_ALARMIN_SHORT_CIRCUIT)*/
    }
```

```

        "inductiveEventType":"",
/*optional, string, inductive event type (only valid for rear-end devices)*/
        "netUser":"",
/*optional, string, user name for network operation*/
        "remoteHostAddr":"",
/*optional, string, remote host address*/
        "cardType":1,
/*optional, int, card type: 1 (normal card), 2 (disability card), 3 (blocklist
card), 4 (patrol card), 5 (duress card), 6 (super card), 7 (visitor card), 8
(dismiss card)*/
        "cardReaderNo":1,
/*optional, int, card reader No.*/
        "doorNo":1,
/*optional, int, door (floor) No.*/
        "deviceNo":1,
/*optional, int, device No.*/
        "serialNo":1,
/*optional, int, event serial No.*/
        "QRCodeInfo":"",
/*optional, string, QR code information*/
        "thermometryUnit":"",
/*optional, string, temperature unit: "celsius", "fahrenheit", "kelvin"*/
        "currTemperature":1.0,
/*optional, float, face temperature which is accurate to one decimal place*/
        "isAbnormalTemperature":true,
/*optional, boolean, whether the face temperature is abnormal: true, false*/
        "RegionCoordinates":{
/*optional, coordinates of the face temperature*/
            "positionX":1,
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
            "positionY":1
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
        },
        "mask":"",
/*optional, string, whether the person is wearing a mask: "unknown", "yes",
"no"*/
        "frontSerialNo":1,
/*optional, int, serial No. of the previous event. If this node is not returned
by the device, the platform will check whether the event is lost by serialNo;
if this node is returned by the device, the platform will check whether the
event is lost by both this node and serialNo. This node is used for the problem
that the serialNo is not continuous after alarm subscription*/
        "IDCardInfo":{
            "name":"",
/*optional, string, name*/
            "sex":"",
/*optional, string, gender: "male", "female"*/
            "birth":"",
/*optional, string, date of birth, e.g., "1990-02-24"*/
            "addr":"",
/*optional, string, address*/
            "IDCardNo":"",

```

```
/*optional, string, ID card No.*/
    "issuingAuthority":"",
/*optional, string, issuing authority*/
    "startDate":"",
/*optional, string, start date of the validity period*/
    "endDate":"",
/*optional, string, end date of the validity period*/
    "isLongTermEffective":false
/*optional, boolean, whether it is permanently valid*/
},
    "capturePicUrl":"",
/*optional, string, captured picture URL*/
    "IDCardPic":"",
/*optional, string, ID card picture URL*/
    "visibleLightPicUrl":"",
/*optional, string, visible light picture URL*/
    "thermalPicUrl":"",
/*optional, string, thermal picture URL*/
    "helmet":"",
/*optional, string, whether the person wears a hard hat: "unknown", "yes",
"no"*/
    "dateTime":"2016-12-12T17:30:08+08:00",
/*required, string, alarm triggering time (UTC time), the maximum size is 32
bytes*/
    "HealthInfo": {
/*optional, object, health information*/
        "healthCode": 1,
/*optional, int, health code status: 0 (no request), 1 (no health code), 2
(green QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6
(other error, e.g., searching failed due to API exception), 7 (searching for
the health code timed out)*/
        "NADCode": 1,
/*optional, int, nucleic acid test result: 0 (no result), 1 (negative, which
means normal), 2 (positive, which means diagnosed), 3 (the result has expired)*/
        "travelCode": 1,
/*optional, int, trip code: 0 (no trip in the past 14 days), 1 (once left in
the past 14 days), 2 (has been to the high-risk area in the past 14 days), 3
(other)*/
        "vaccineStatus": 1
/*optional, int, whether the person is vaccinated: 0 (not vaccinated), 1
(vaccinated)*/
    }
}
}
```

## See Also

### [Access Control Event Types](#)

## F.123 JSON\_IDCardInfoEventCap

JSON message about the capability of getting the ID card swiping events actively

```
{  
    "IDCardInfoEventCap":{  
        "IDCardInfoEventCond":{  
            "searchID":{  
                /*required, string, search ID, which is used to check whether the current  
                search requester is the same as the previous one. If they are the same, the  
                search record will be stored in the device to speed up the next search*/  
                "@min":1,  
                "@max":1  
            },  
            "searchResultPosition":{  
                /*required, int, the end position of search result in result list. In a single  
                search, if you cannot get all the records in the result list, you can mark the  
                end position and get the following records after the marked position in the  
                next search. For example, if the maximum value of totalMatches supported by the  
                device is M, but there are N matched results stored in the device currently  
                (N<=M), the valid range of this node is 0 to N-1*/  
                "@min":1,  
                "@max":1  
            },  
            "maxResults":{  
                /*required, int, maximum number of records that can be obtained after the URI  
                is called this time. If the value of maxResults is larger than the value  
                returned by the device capability, the device will return according to the  
                maximum value in the capability and will not return error information*/  
                "@min":1,  
                "@max":1  
            },  
            "major":{  
                /*required, int, major event type: 0-all, 1-alarm, 2-exception, 3-operaiton, 5-  
                event. For details, refer to Access Control Event Types. The value of this node  
                is in decimal format instead of hexadecimal format*/  
                "@opt":[0, 1, 2, 3, 5]  
            },  
            "minorAlarm":{  
                /*required, int, minor alarm type. For details, refer to Access Control Event  
                Types. The value of this node is in decimal format instead of hexadecimal  
                format*/  
                "@opt":[1024, 1025, 1026, 1027]  
            },  
            "minorException":{  
                /*required, int, minor exception type. For details, refer to Access Control  
                Event Types. The value of this node is in decimal format instead of hexadecimal  
                format*/  
                "@opt":[39, 58, 59, 1024]  
            },  
            "minorOperation":{  
                /*required, int, minor operation type. For details, refer to Access Control  
                Event Types. The value of this node is in decimal format instead of hexadecimal  
                format*/  
                "@opt":[39, 58, 59, 1024]  
            }  
        }  
    }  
}
```

```
/*required, int, minor operation type. For details, refer to Access Control Event Types. The value of this node is in decimal format instead of hexadecimal format*/
    "@opt": [80, 90, 112, 113]
},
"minorEvent":{
/*required, int, minor event type. For details, refer to Access Control Event Types. The value of this node is in decimal format instead of hexadecimal format*/
    "@opt": [1, 2, 3, 4]
},
"startTime": "1970-01-01T00:00:00+00:00",
/*optional, string, start time (UTC time)*/
    "endTime": "2017-12-12T17:30:08+08:00",
/*optional, string, end time (UTC time)*/
    "picEnable":{
/*optional, boolean, whether to upload events with pictures: true (yes), false (no). The default value is true*/
        "@opt": [true, false]
},
"beginSerialNo":{
/*optional, int, start serial No.*/
    "@min": 1,
    "@max": 1
},
"endSerialNo":{
/*optional, int, end serial No.*/
    "@min": 1,
    "@max": 1
},
    "isAbnormalTemperature": {
/*optional, object, whether the skin-surface temperature is abnormal*/
        "@opt": [true, false]
    },
/*optional, array of boolean, options: true (yes), false (no)*/
    }
},
"InfoList": {
/*optional, event information*/
    "deviceName": {
/*optional, string, device name*/
        "@min": 1,
        "@max": 1
    },
    "inductiveEventType": {
/*optional, string, inductive event type (only valid for rear-end devices)*/
        "@min": 1,
        "@max": 1
    },
    "netUser": {
/*optional, string, user name for network operation*/
        "@min": 1,
        "@max": 1
    }
}
```

```
},
    "remoteHostAddr":{
/*optional, string, remote host address*/
        "@min":1,
        "@max":1
    },
    "cardType":{
/*optional, int, card type: 1 (normal card), 2 (disability card), 3 (blocklist
card), 4 (patrol card), 5 (duress card), 6 (super card), 7 (visitor card), 8
(dismiss card)*/
        "@opt":[1, 2, 3, 4, 5, 6, 7, 8]
    },
    "cardReaderNo":{
/*optional, int, card reader No.*/
        "@min":1,
        "@max":1
    },
    "doorNo":{
/*optional, int, door (floor) No.*/
        "@min":1,
        "@max":1
    },
    "deviceNo":{
/*optional, int, device No.*/
        "@min":1,
        "@max":1
    },
    "serialNo":{
/*optional, int, event serial No.*/
        "@min":1,
        "@max":1
    },
    "QRCodeInfo":{
/*optional, string, QR code information*/
        "@min":1,
        "@max":1
    },
    "thermometryUnit":{
/*optional, string, temperature unit: "celsius", "fahrenheit", "kelvin"*/
        "@opt":["celsius", "fahrenheit", "kelvin"]
    },
    "currTemperature":{
/*optional, float, face temperature which is accurate to one decimal place*/
        "@min":1.0,
        "@max":1.0
    },
    "isAbnormalTemperature":{
/*optional, boolean, whether the face temperature is abnormal: true, false*/
        "@opt":[true, false]
    },
    "RegionCoordinates":{
/*optional, coordinates of the face temperature*/
```

```

        "positionX": {
/*optional, int, normalized X-coordinate which is between 0 and 1000*/
            "@min":1,
            "@max":1
        },
        "positionY": {
/*optional, int, normalized Y-coordinate which is between 0 and 1000*/
            "@min":1,
            "@max":1
        }
    },
    "mask": {
/*optional, string, whether the person is wearing a mask: "unknown", "yes",
"no"*/
        "@opt": ["unknown", "yes", "no"]
    },
    "frontSerialNo": {
/*optional, int, serial No. of the previous event. If this node is not returned
by the device, the platform will check whether the event is lost by serialNo;
if this node is returned by the device, the platform will check whether the
event is lost by both this node and serialNo. This node is used for the problem
that the serialNo is not continuous after alarm subscription*/
        "@min":1,
        "@max":1
    },
    "IDCardInfo": {
        "name": {
/*optional, string, name*/
            "@min":1,
            "@max":1
        },
        "sex": {
/*optional, string, gender: "male", "female"*/
            "@min":1,
            "@max":1
        },
        "birth": {
/*optional, string, date of birth, e.g., "1990-02-24"*/
            "@min":1,
            "@max":1
        },
        "addr": {
/*optional, string, address*/
            "@min":1,
            "@max":1
        },
        "IDCardNo": {
/*optional, string, ID card No.*/
            "@min":1,
            "@max":1
        },
        "issuingAuthority": {

```

```

/*optional, string, issuing authority*/
    "@min":1,
    "@max":1
},
"startDate":{
/*optional, string, start date of the validity period*/
    "@min":1,
    "@max":1
},
"endDate":{
/*optional, string, end date of the validity period*/
    "@min":1,
    "@max":1
},
"isLongTermEffective":{
/*optional, boolean, whether it is permanently valid*/
    "@opt":[true, false]
}
},
"capturePicUrl":{
/*optional, string, captured picture URL*/
    "@min":1,
    "@max":1
},
"IDCardPic":{
/*optional, string, ID card picture URL*/
    "@min":1,
    "@max":1
},
"visibleLightPicUrl":{
/*optional, string, visible light picture URL*/
    "@min":1,
    "@max":1
},
"thermalPicUrl":{
/*optional, string, thermal picture URL*/
    "@min":1,
    "@max":1
},
"helmet":{
/*optional, string, whether the person wears a hard hat: "unknown", "yes",
"no"*/
    "@opt":["unknown", "yes", "no"]
},
"dateTime":"2016-12-12T17:30:08+08:00",
/*required, string, alarm triggering time (UTC time), the maximum size is 32
bytes*/
"HealthInfo":{
/*optional, object, health information*/
    "healthCode":{
/*optional, object, health code status*/
        "@opt":[0, 1, 2, 3, 4, 5, 6]
}
}
}

```

```
/*optional, array of int, options: 0 (no request), 1 (no health code), 2 (green
QR code), 3 (yellow QR code), 4 (red QR code), 5 (no such person), 6 (other
error, e.g., searching failed due to API exception), 7 (searching for the
health code timed out)*/
    },
    "NADCode":{

/*optional, object, nucleic acid test result: 0 (no result), 1 (negative, which
means normal), 2 (positive, which means diagnosed), 3 (the result has expired)*/
        "@opt":[0, 1, 2, 3]
    },
    "travelCode":{

/*optional, object, trip code: 0 (no trip in the past 14 days), 1 (once left in
the past 14 days), 2 (has been to the high-risk area in the past 14 days), 3
(other)*/
        "@opt":[0, 1, 2, 3]
    },
    "vaccineStatus":{

/*optional, object, whether the person is vaccinated: 0 (not vaccinated), 1
(vaccinated)*/
        "@opt":[0, 1]
    }
}
}
```

## See Also

## Access Control Event Types

## F.124 JSON\_IDCardInfoEventCond

JSON message about the condition of getting ID card swiping events actively

```
{  
    "IDCardInfoEventCond": {  
        "searchID": "",  
        /*required, string, search ID, which is used to check whether the current  
        search requester is the same as the previous one. If they are the same, the  
        search record will be stored in the device to speed up the next search*/  
        "searchResultPosition": 0,  
        /*required, int, the end position of search result in result list. In a single  
        search, if you cannot get all the records in the result list, you can mark the  
        end position and get the following records after the marked position in the  
        next search. For example, if the maximum value of totalMatches supported by the  
        device is M, but there are N matched results stored in the device currently  
        (N<=M), the valid range of this node is 0 to N-1*/  
        "maxResults": 30,  
        /*required, int, maximum number of records that can be obtained after the URI  
        is called this time. If the value of maxResults is larger than the value  
        returned by the device capability, the device will return according to the
```

```
maximum value in the capability and will not return error information*/
    "major":1,
/*optional, int, major event type, 0 means all event types. For details, refer
to Access Control Event Types. The value of this node is in decimal format
instead of hexadecimal format (for example, 1 refers to 0x1 which indicates
that the major type is MAJOR_ALARM)*/
    "minor":1024,
/*optional, int, minor event type, 0 means all event types. For details, refer
to Access Control Event Types. The value of this node is in decimal format
instead of hexadecimal format (for example, 1024 refers to 0x400 which
indicates that the minor type is MINOR_ALARMIN_SHORT_CIRCUIT)*/
    "startTime":"2016-12-12T17:30:08+08:00",
/*optional, string, start time (UTC time)*/
    "endTime":"2017-12-12T17:30:08+08:00",
/*optional, string, end time (UTC time)*/
    "picEnable":true,
/*optional, boolean, whether to upload events with pictures: true (yes), false
(no). The default value is true*/
    "beginSerialNo":1,
/*optional, int, start serial No.*/
    "endSerialNo":1,
/*optional, int, end serial No.*/
    "isAbnormalTemperature":true
/*optional, boolean, whether the skin-surface temperature is abnormal*/
}
}
```

## See Also

[\*\*Access Control Event Types\*\*](#)

## F.125 JSON\_IdentityInfo

IdentityInfo message in JSON format

```
{
    "IdentityInfo":{
        "chnName":"",
/*optional, string, reserved*/
        "enName":"",
/*optional, string, English name*/
        "sex":"",
/*optional, string, gender: "male", "female"*/
        "birth":"",
/*optional, string, date of birth, e.g., 1990-02-24*/
        "addr":"",
/*optional, string, address*/
        "IDCardNo":"",
/*optional, string, ID card No., it is the sensitive information that should be
encrypted*/
        "issuingAuthority":"",

```

```

/*optional, string, authority*/
    "startDate":"",
/*optional, string, start time of the validity period*/
    "endDate":"",
/*optional, string, end time of the validity period*/
    "passNo":"",
/*optional, string, entry-exit permit No.*/
    "issueNumber":"",
/*optional, string, issuing times*/
    "certificateType":"",
/*optional, string, certificate type*/
    "permanentResidenceCardNo":"",
/*optional, string, permanent resident card No.*/
    "nationalityOrAreaCode":"",
/*optional, string, country or region code*/
    "version":"",
/*optional, string, certificate version No.*/
    "receivingAuthorityCode":"",
/*optional, string, acceptance authority code*/
    "FingerprintList":[{
        "fingerprint":""
    }],
/*optional, string, fingerprint information, it is encoded using base64*/
    "pic":""
/*optional, string, ID photo information, it is encoded using base64. The
encrypted data should be decrypted using the specific decryption library*/
}
}

```

### F.126 JSON\_IdentityInfoCap

IdentityInfoCap capability message in JSON format

```

{
    "IdentityInfoCap": {
        "IdentityInfoCond": {
        },
/*optional, conditions of collecting ID card information*/
        "chnName": {
            "@min": 0,
            "@max": 0
        },
        "enName": {
            "@min": 0,
            "@max": 0
        },
        "sex": {
            "@opt": ["male", "female"]
        }
    }
}

```

```
},
"birth": {
    /*optional, string, date of birth, e.g., 1990-02-24*/
    "@min":0,
    "@max":0
},
"addr": {
    /*optional, string, address*/
    "@min":0,
    "@max":0
},
"IDCardNo": {
    /*optional, string, ID card No.*/
    "@min":0,
    "@max":0
},
"issuingAuthority": {
    /*optional, string, authority*/
    "@min":0,
    "@max":0
},
"startDate": {
    /*optional, string, start time of the validity period*/
    "@min":0,
    "@max":0
},
"endDate": {
    /*optional, string, end time of the validity period*/
    "@min":0,
    "@max":0
},
"passNo": {
    /*optional, string, entry-exit permit No.*/
    "@min":0,
    "@max":0
},
"issueNumber": {
    /*optional, string, issuing times*/
    "@min":0,
    "@max":0
},
"certificateType": {
    /*optional, string, certificate type*/
    "@min":0,
    "@max":0
},
"permanentResidenceCardNo": {
    /*optional, string, permanent resident card No.*/
    "@min":0,
    "@max":0
},
"nationalityOrAreaCode": {
```

```
/*optional, string, country or region code*/
    "@min":0,
    "@max":0
},
"version":{
/*optional, string, certificate version No.*/
    "@min":0,
    "@max":0
},
"receivingAuthorityCode":{
/*optional, string, acceptance authority code*/
    "@min":0,
    "@max":0
},
"FingerprintList":{
    "maxSize":0,
    "fingerprint":{
/*optional, string, fingerprint information, it is encoded using base64. This
field is the data size capability*/
        "@min":0,
        "@max":0
    }
},
"pic":{
/*optional, string, ID photo information, it is encoded using base64. This
field is the data size capability*/
    "@min":0,
    "@max":0
}
}
```

### F.127 JSON\_IdentityInfoCond

IdentityInfoCond message in JSON format

```
{
    "IdentityInfoCond":{ }
/*currently there are no condition parameters, so this field can be set to
NULL*/
}
```

### F.128 JSON\_IRCfg

JSON message about active infrared intrusion parameters

```
{
    "IRCfg": {
        "enable": ,
```

```
/*required, boolean, whether to enable: true (yes), false (no)*/
    "distance":  
/*optional, float, distance, unit: m*/
    }  
}
```

### F.129 JSON\_IRCfgCap

JSON message about active infrared intrusion capability

```
{
  "IRCfgCap": {
    "enable": [true, false],
    /*required, boolean, whether to enable*/
    "distance": {
      "@opt": [0.5, 1, 1.5]
    }
  }
}
```

### F.130 JSON\_LogModeCfg

LogModeCfg message in JSON format

```
{
  "LogModeCfg": {
    "type":  
/*optional, integer, log mode: 1-16 bytes (the host log can be stored by 25w,  
and the employee No. can be stored by 16 bytes), 2-12 bytes (the host log can  
be stored by 25w, and the employee No. can be stored by 12 bytes). This node  
will be set to 1 by default*/
  }
}
```

### F.131 JSON\_MaskDetection

Message about the mask detection parameters in JSON format.

```
{
  "MaskDetection": {
    "enable": ,
    /*optional, boolean, whether to enable mask detection: true-enable, false-
    disable*/
    "noMaskStrategy": ""
    /*optional, string, door control strategy when not wearing mask is detected:  
"noTipsAndOpenDoor"-open the door without prompt, "tipsAndOpenDoor"-prompt and  
open the door (default), "tipsAndNotOpenDoor"-prompt and not open the door.
  }
}
```

```
This field is valid when enable is true*/  
}  
}
```

### F.132 JSON\_MaskDetectionCap

Message about the configuration capability of mask detection in JSON format.

```
{  
    "MaskDetectionCap":{  
        "enable":{  
            /*optional, boolean, whether to enable mask detection: true-enable, false-disable*/  
            "@opt":[true, false]  
        },  
        "noMaskStrategy":{  
            /*optional, string, door control strategy when not wearing mask is detected:  
            "noTipsAndOpenDoor"-open the door without prompt, "tipsAndOpenDoor"-prompt and  
            open the door (default), "tipsAndNotOpenDoor"-prompt and not open the door.  
            This field is valid when enable is true*/  
            "@opt":["noTipsAndOpenDoor", "tipsAndOpenDoor", "tipsAndNotOpenDoor"]  
        }  
    }  
}
```

### F.133 JSON\_NFCCfg

NFCCfg message in JSON format

```
{  
    "NFCCfg":{  
        "enable":  
            /*required, boolean, whether to enable NFC function: "true"-yes, "false"-no*/  
    }  
}
```

### F.134 JSON\_NFCCfgCap

NFCCfgCap capability message in JSON format

```
{  
    "NFCCfgCap":{  
        "enable":"true, false"  
        /*required, whether to enable NFC function: "true"-yes, "false"-no (default)*/  
    }  
}
```

## F.135 JSON\_OfflineCaptureCap

OfflineCaptureCap capability message in JSON format

```
{
    "OfflineCaptureCap": {
        "isSupportDownloadOfflineCaptureInfoTemplate": true,
        /*optional, whether it supports downloading template of offline user list:
        "true"-yes, this node is not returned-no*/
        "isSupportUploadOfflineCaptureInfo": true,
        /*optional, whether it supports uploading offline user list: "true"-yes, this
        node is not returned-no*/
        "isSupportDownloadCaptureData": true,
        /*optional, whether it supports downloading collected data: "true"-yes, this
        node is not returned-no*/
        "isSupportDeleteAllData": true,
        /*optional, whether it supports deleting all collected data: "true"-yes, this
        node is not returned-no*/
        "isSupportDeleteTheData": true,
        /*optional, whether it supports deleting specific collected data: "true"-yes,
        this node is not returned-no*/
        "SearchTask": {
            "supportFunction": {
                /*required, string, supported methods, actually supported methods will be
                returned*/
                "@opt": ["put", "get", "delete", "post"]
            },
            "searchID": {
                /*required, string, search ID which is used to check whether the upper-layer
                clients are the same one*/
                "@min": 0,
                "@max": 0
            },
            "maxResults": {
                "@min": 0,
                "@max": 0
            },
            "captureNoList": {
                "maxSize": 0,
                "@min": 0,
                "@max": 0
            },
            "searchType": {
                "@opt": ["new", "modified"]
            },
            "DataCollections": {
                /*optional, array, matched data information that has been searched*/
                "maxSize": 0,
                "captureNo": {
                    /*optional, integer, collection No.*/
                    "@min": 0,
                    "@max": 0
                }
            }
        }
    }
}
```

```

        "@max":0
    },
    "name":{
/*optional, string, name*/
        "@min":0,
        "@max":0
    },
    "employeeNo":{
/*optional, string, employee No.*/
        "@min":0,
        "@max":0
    },
    "CardNoList":{
/*optional, string, card No. list*/
        "maxSize":0,
        "cardNo":{
            "@min": 0,
            "@max": 0
        },
        "cardType": {
/*optional, string, card type: "TypeA_M1", "TypeA_CPU", "TypeB", "ID_125K",
"FelicaCard", "DesfireCard"*/
            "@opt":
["TypeA_M1","TypeA_CPU","TypeB","ID_125K","FelicaCard","DesfireCard"]
        }
    },
    "IDCardNo":{
/*optional, string, ID card No.*/
        "@min":0,
        "@max":0
    },
    "FingerprintList":{
        "fingerprintID":{
            "@min":0,
            "@max":0
        },
        "fingerprint":{
/*optional, fingerprint information, it is encoded using base64*/
            "@min":0,
            "@max":0
        }
    },
    "FaceFeature":{
/*optional, string, facial feature information*/
        "isSupportFaceRegion":true,
/*optional, whether it supports facial feature area*/
        "isSupportCommonPoint":true
/*optional, whether it supports feature point coordinates (e.g., left eye,
right eye, left mouth corner, right mouth corner, nose)*/
    },
    "isSupportRiskMark":true,
/*optional, whether it supports risk data mark*/

```

```
        "dataType": {
/*optional, data type*/
            "@opt": ["new", "modified", "normal"]
        },
        "IdentityInfo": {
/*identity information*/
            "chnName": {
/*optional, string, Chinese name*/
                "@min":0,
                "@max":0
            },
            "enName": {
/*optional, string, English name*/
                "@min":0,
                "@max":0
            },
            "sex": {
/*optional, string, gender: "male", "female"*/
                "@opt": ["male", "female"]
            },
            "birth": {
/*optional, string, data of birth, e.g., "1990-02-24"*/
                "@min":0,
                "@max":0
            },
            "addr": {
/*optional, string, address*/
                "@min":0,
                "@max":0
            },
            "IDCardNo": {
/*optional, string, ID card No.*/
                "@min":0,
                "@max":0
            },
            "issuingAuthority": {
/*optional, string, issuing authority*/
                "@min":0,
                "@max":0
            },
            "startDate": {
/*optional, string, start date of validity period*/
                "@min":0,
                "@max":0
            },
            "endDate": {
/*optional, string, end date of validity period*/
                "@min":0,
                "@max":0
            },
            "passNo": {
/*optional, string, entry-exit permit No.*/

```

```

        "@min":0,
        "@max":0
    },
    "issueNumber":{
/*optional, string, issued times*/
        "@min":0,
        "@max":0
    },
    "certificateType":{
/*optional, string, certificate type*/
        "@min":0,
        "@max":0
    },
    "permanentResidenceCardNo":{
/*optional, string, permanent resident visa No.*/
        "@min":0,
        "@max":0
    },
    "nationalityOrAreaCode":{
/*optional, string, country/region code*/
        "@min":0,
        "@max":0
    },
    "version":{
/*optional, string, certificate version No.*/
        "@min":0,
        "@max":0
    },
    "receivingAuthorityCode":{
/*optional, string, acceptance authority code*/
        "@min":0,
        "@max":0
    },
    "FingerprintList":{
        "maxSize":0,
        "fingerprint":{
/*optional, string, fingerprint information, which should be encoded by Base64*/
            "@min":0,
            "@max":0
        }
    },
    "pic":{
/*optional, string, certificate picture information, which should be encoded by
Base64, encrypted and decrypted by a specific decryption library*/
        "@min":0,
        "@max":0
    }
},
    "CardIssueStatus":{
/*optional, issuing status list of cards containing face pictures and
fingerprints*/
        "@size":0,

```

```

/*optional, capability of number of elements in the array*/
    "face":{
/*optional, boolean, card issuing status of the face picture: true-with card
issued, false-without card issued*/
        "@opt":[true, false]
    },
    "fingerprint1":{
/*optional, boolean, card issuing status of the fingerprint 1: true-with card
issued, false-without card issued*/
        "@opt":[true, false]
    },
    "fingerprint2":{
/*optional, boolean, card issuing status of the fingerprint 2: true-with card
issued, false-without card issued*/
        "@opt":[true, false]
    }
},
"RuleInfo":{
/*rule list, which lists rules for collecting different types of data*/
    "reqAdminRights":[true, false],
/*required, boolean, whether the administrator permission is required: "true"-yes,
 "false"-no*/
    "enableCardNoLenAuto":[true, false],
/*optional, boolean, whether to enable length self-adaption of the card serial
No.*/
    "maxSize":0,
    "supportFunction":{
/*required, string, supported methods, actually supported methods will be
returned*/
        "@opt":["put", "get", "delete", "post"]
    },
    "dataType":{
/*required, string, data type: "name", "employeeNo"-employee No., "IDCardNo"-ID
card No., "IDCardSerialNo"-ID card serial No., "IDCardDetails"-ID card
details, "card", "fingerprint"-fingerprint, "face"*/
        "@opt":["name", "employeeNo", "IDCardNo", "IDCardSerialNo",
"IDCardDetails", "card", "fingerprint", "face"]
    },
    "enable":[true, false],
/*required, string, whether to collect and display: "true"-collect and display,
"false"-not collect and display*/
    "uniqueCheck":[true, false],
/*dependency, boolean, whether to enable uniqueness verification: "true"-yes,
"false" (default) or this node is not returned-no. This field is valid when
dataType is "name". For other data types, the field is the read-only optional
parameter*/
    "len":{

/*dependency, integer, data length. If dataType is "name", it refers to the
name length and the default value is 128. For other data types, this field is
the read-only optional parameter. This node will not be returned if it is not

```

```

supported. The capability list will be returned according to the data type*/
    "dataType":"",
    "@min":0,
    "@max":0
},
"num":{

/*dependency, integer, number of collected data, this field is valid when
dataType is "fingerprint" or "card". The capability list will be returned
according to the data type*/
    "dataType":"",
    "@min":0,
    "@max":0
},
"fingerprintIDs":{

/*dependency, integer, No. list of collected fingerprints, this field is valid
when dataType is "fingerprint"*/
    "maxSize":0,
    "@min":0,
    "@max":0
},
"enableLocalIssueCard": {
/*optional, boolean, whether to enable issuing smart cards locally*/
    "@opt": [true,false]
},
"isLocalStorage": {

/*optional, boolean, whether to store face picture and fingerprint information
in the device locally*/
    "@opt": [true,false]
}
},
"CaptureProgress":{

"supportFunction":{

/*required, string, supported methods, actually supported methods will be
returned*/
    "@opt":["put", "get", "delete", "post"]
},
"reqCaptureNum":{

/*optional, integer, total number of persons to be collected*/
    "@min":0,
    "@max":0
},
"completelyCaptureNum":{

/*optional, integer, number of completely collected persons*/
    "@min":0,
    "@max":0
},
"partiallyCaptureNum":{

/*optional, integer, number of partially collected persons*/
    "@min":0,
    "@max":0
},
"reqFaceNum":{

```

```
/*optional, integer, number of faces to be collected*/
    "@min":0,
    "@max":0
},
"faceNum":{

/*optional, integer, number of collected faces*/
    "@min":0,
    "@max":0
},
"reqFingerprintNum":{

/*optional, integer, number of fingerprints to be collected*/
    "@min":0,
    "@max":0
},
"fingerprintNum":{

/*optional, integer, number of collected fingerprints*/
    "@min":0,
    "@max":0
},
"reqCardNum":{

/*optional, integer, number of cards to be collected*/
    "@min":0,
    "@max":0
},
"cardNum":{

/*optional, integer, number of collected cards*/
    "@min":0,
    "@max":0
},
"reqIDCardNum":{

/*optional, integer, number of ID cards to be collected*/
    "@min":0,
    "@max":0
},
"IDCardNum":{

/*optional, integer, number of collected ID cards*/
    "@min":0,
    "@max":0
},
"reqIssueNum":{

/*optional, int, number of persons to be issued with smart cards*/
    "@min": 0,
    "@max": 0
},
"IssuedNum":{

/*optional, int, number of persons that have been issued with smart cards*/
    "@min": 0,
    "@max": 0
},
"DataOutput":{

    "supportFunction":{
```

```
/*required, string, supported methods, actually supported methods will be
returned*/
    "@opt": ["put", "get", "delete", "post"]
},
"password": {
/*required, string, password for exporting*/
    "@min": 0,
    "@max": 0
},
"type": {
/*optional, string, exporting method, the default method is "USB"*/
    "@opt": "USB"
},
"progress": {
/*required, integer, exporting progress*/
    "@min": 0,
    "@max": 0
}
}
}
```

### F.136 JSON\_OSDPModify

OSDPModify message in JSON format

```
{
  "OSDPModify": {
    "newID": {
/*required, integer, new ID of the OSDP card reader*/
    }
  }
}
```

### F.137 JSON\_OSDPStatus

OSDPStatus message in JSON format

```
{
  "OSDPStatus": {
    "status": ""
/*required, string, online status: "online", "offline"*/
  }
}
```

## F.138 JSON\_PersonInfoExtendName

JSON message about the parameters of the name of the additional person information

```
{  
    "PersonInfoExtendName": {  
        "NameList": [{  
            "id": 1,  
            /* required, int, ID of the additional person information, it corresponds to the  
            id of PersonInfoExtends in the message JSON_UserInfo */  
            "name": "Student ID"  
            /* required, string, name of the additional person information */  
        }]  
    }  
}
```

### See Also

[JSON\\_UserInfo](#)

## F.139 JSON\_PersonInfoExtendNameCap

JSON message about the configuration capability of the name of the additional person information

```
{  
    "PersonInfoExtendNameCap": {  
        "NameList": {  
            "@size": 1,  
            /* required, int, maximum number of names that can be configured */  
            "id": {  
                /* required, int, ID of the additional person information, it corresponds to the  
                id of PersonInfoExtends in the message JSON_Cap_UserInfo */  
                "@min": 1,  
                "@max": 1  
            },  
            "name": {  
                /* required, string, name of the additional person information */  
                "@min": 1,  
                "@max": 1  
            }  
        }  
    }  
}
```

### See Also

[JSON\\_Cap\\_UserInfo](#)

## F.140 JSON\_PictureServerInformation

PictureServerInformation message in JSON format

```
{
    "PictureServerInformation": {
        "pictureServerType": "",
        /*required, string type, picture storage server type:
        "tomact,VRB,cloudStorage,KMS"*/
        "addressingFormatType": "",
        /*required, string type, format type of the picture storage server address:
        "ipaddress"-IP address (default), "hostname"-host name*/
        "hostName": "",
        /*string type, domain name of the picture storage server, the string length is
        between 0 and 64. This field is valid when addressingFormatType is "hostname"*/
        "ipv4Address": "",
        /*string type, IPv4 address of the picture storage server, the string length is
        between 0 and 64. This field is valid when addressingFormatType is "ipaddress"*/
        "ipv6Address": "",
        /*string type, IPv6 address of the picture storage server, the string length is
        between 0 and 128. This field is valid when addressingFormatType is
        "ipaddress"*/
        "portNo": ,
        /*required, integer type, port No. of the picture storage server, which is
        between 1024 and 65535*/
        "underlyingProtocol": "",
        /*optional, string, bottom-level protocol of the picture storage server:
        "HTTP", "HTTPS". This field is valid when pictureServerType contains
        "cloudStorage". If this field does not exist, the default bottom-level protocol
        is HTTP*/
        "cloudStorage": {
            /*parameters of the cloud storage server, which is valid when
            pictureServerType is "cloudStorage"*/
            "cloudManageHttpPort": ,
            /*required, integer type, HTTP port No. for central management of the cloud
            storage server, which is between 1024 and 65535*/
            "cloudTransDataPort": ,
            /*required, integer type, data transmission port No. of the cloud storage
            server, which is between 1024 and 65535. This field is not supported by access
            control devices*/
            "cloudCmdPort": ,
            /*required, integer type, signaling port No. of the cloud storage server, which
            is between 1024 and 65535*/
            "cloudHeartBeatPort": ,
            /*required, integer type, heartbeat port No. of the cloud storage server, which
            is between 1024 and 65535. This field is not supported by access control
            devices*/
            "cloudStorageHttpPort": ,
            /*required, integer type, HTTP port No. of the cloud storage server, which is
            between 1024 and 65535. This field is not supported by access control devices*/
            "cloudUsername": ""
        }
    }
}
```

```
/*required, string type, user name of the cloud storage server, the string length is between 0 and 32. This field is not supported by access control devices*/
    "cloudPassword": "",
/*required, string type, password of the cloud storage server, the string length is between 0 and 32. This field is not supported by access control devices*/
    "cloudPoolId": ,
/*required, integer type, cloud storage pool ID, which is between 1 and 4294967295. If this field is not configured by the upper-level, this field will be set to 1 by default*/
    "cloudPoolIdEx": "",
/*optional, string type, cloud storage pool ID, this node is valid when cloud storage pool ID of type string (cloud storage protocol in version 3.0) is supported*/
    "cloudProtocolVersion": "",
/*required, string type, protocol version of the cloud storage server, the string length is between 0 and 32*/
    "cloudAccessKey": "",
/*string type, cloud storage server access_key, the string length is between 0 and 64. This field is valid when cloudProtocolVersion is "V2.0"*/
    "cloudSecretKey": ""
/*string type, cloud storage server secret_key, the string length is between 0 and 64. This field is valid when cloudProtocolVersion is "V2.0"*/
}
}
```

### F.141 JSON\_QRCodeEvent

JSON message about the result of actively getting QR code scanning events

```
{
  "QRCodeEvent" : {
    "searchID": "",
/*required, string, search ID, which is used to check whether the current search requester is the same as the previous one. If they are the same, the search record will be stored in the device to speed up the next search*/
    "responseStatusStrg": "OK",
/*required, string, search status: "OK"(searching completed), "MORE"(searching for more results), "NO MATCH"(no matched results)*/
    "numOfMatches": 1,
/*required, int, the number of the returned records*/
    "totalMatches": 1,
/*required, int, the total number of the matched records*/
    "InfoList" : [
/*optional, event information*/
      "deviceName": "",
/*optional, string, device name*/
      "serialNo": 1,
```

```

/*optional, int, event serial No.*/
    "QRCodeInfo": "",
/*required, string, QR code information*/
    "thermometryUnit": "",
/*optional, string, temperature unit: "celsius" (Celsius), "fahrenheit"
(Fahrenheit), "kelvin" (Kelvin), the default value is "celsius"*/
    "currTemperature": 1.0,
/*optional, float, face temperature, the value is accurate to one decimal
place*/
    "isAbnormalTemperature": true,
/*optional, boolean, whether the face temperature is abnormal: true (abnormal),
false (normal)*/
    "RegionCoordinates": {
/*optional, coordinates of the face temperature*/
        "positionX": 1,
/*optional, int, X-coordinate, the value is normalized to a number between 0
and 1000*/
        "positionY": 1
/*optional, int, Y-coordinate, the value is normalized to a number between 0
and 1000*/
    },
    "mask": "",
/*optional, string, whether the person wears a mask: "unknown" (unknown), "yes"
(wearing a mask), "no" (no mask)*/
    "visibleLightPicUrl":"",
/*optional, string, the URL of the visible light picture*/
    "thermalPicUrl":"",
/*optional, string, the URL of the thermal picture*/
    "helmet": "",
/*optional, string, whether the person wears a hard hat: "unknown" (unknown),
"yes" (wearing a hard hat), "no" (no hard hat)*/
    "dateTime": "2016-12-12T17:30:08+08:00"
/*required, string, the time (UTC time) when the alarm is triggered, the
maximum size is 32 bytes*/
}
}
}

```

### F.142 JSON\_QRCodeEventCap

JSON message about the capability of actively getting QR code scanning events

```

{
    "QRCodeEventCap": {
        "QRCodeEventCond": {
            "searchID": {
/*required, string, search ID, which is used to check whether the current
search requester is the same as the previous one. If they are the same, the
search record will be stored in the device to speed up the next search*/
                "@min":1,

```

```

        "@max":1
    },
    "searchResultPosition": {
/*required, int, the start position of search result in the result list. In a
single search, if you cannot get all the records in the result list, you can
mark the end position and get the following records after the marked position
in the next search. If the maximum number of totalMatches supported by the
device is M and the number of totalMatches stored in the device now is N
(N<=M), the valid range of this node is 0 to N-1*/
        "@min":1,
        "@max":1
    },
    "maxResults": {
/*required, int, the maximum number of search results that can be obtained by
calling the URI this time. If the value of maxResults is greater than that
defined in the device capability, the value in the capability will be returned.
In this case, the device will not return error*/
        "@min":1,
        "@max":1
    },
    "startTime": "1970-01-01T00:00:00+00:00",
/*optional, string, start time (UTC time)*/
    "endTime": "2017-12-12T17:30:08+08:00",
/*optional, string, end time (UTC time)*/
    "picEnable": {
/*optional, boolean, whether to upload the picture along with the event
information: true (all matched events will be uploaded with pictures if there
are any), false (all matched events will be uploaded without pictures). If this
node is not configured, the default value is true*/
        "@opt": [true, false]
    },
    "beginSerialNo": {
/*optional, int, start serial No.*/
        "@min":1,
        "@max":1
    },
    "endSerialNo": {
/*optional, int, end serial No.*/
        "@min":1,
        "@max":1
    }
},
    "InfoList" : {
/*optional, event information*/
        "deviceName": {
/*optional, string, device name*/
            "@min":1,
            "@max":1
        },
        "serialNo": {
/*optional, int, event serial No.*/
            "@min":1,
            "@max":1
        }
    }
}

```

```

        "@max":1
    },
    "QRCodeInfo": {
/*required, string, QR code information*/
        "@min":1,
        "@max":1
    },
    "thermometryUnit": {
/*optional, string, temperature unit: "celsius" (Celsius), "fahrenheit"
(Fahrenheit), "kelvin" (Kelvin), the default value is "celsius"*/
        "@opt":["celsius","fahrenheit","kelvin"]
    },
    "currTemperature": {
/*optional, float, face temperature, the value is accurate to one decimal
place*/
        "@min":1.0,
        "@max":1.0
    },
    "isAbnormalTemperature": {
/*optional, boolean, whether the face temperature is abnormal: true (abnormal),
false (normal)*/
        "@opt":[true,false]
    },
    "RegionCoordinates": {
/*optional, coordinates of the face temperature*/
        "positionX": {
/*optional, int, X-coordinate, the value is normalized to a number between 0
and 1000*/
            "@min":1,
            "@max":1
        },
        "positionY": {
/*optional, int, Y-coordinate, the value is normalized to a number between 0
and 1000*/
            "@min":1,
            "@max":1
        }
    },
    "mask": {
/*optional, string, whether the person wears a mask: "unknown" (unknown), "yes"
(wearing a mask), "no" (no mask)*/
        "@opt":["unknown","yes","no"]
    },
    "visibleLightPicUrl":{
/*optional, string, the URL of the visible light picture*/
        "@min":1,
        "@max":1
    },
    "thermalPicUrl":{
/*optional, string, the URL of the thermal picture*/
        "@min":1,
        "@max":1
    }
}

```

```

        },
        "helmet": {
/*optional, string, whether the person wears a hard hat: "unknown" (unknown),
"yes" (wearing a hard hat), "no" (no hard hat)*/
            "@opt": ["unknown","yes","no"]
        },
        "dateTime": "2016-12-12T17:30:08+08:00"
/*required, string, the time (UTC time) when the alarm is triggered, the
maximum size is 32 bytes*/
    }
}
}

```

### F.143 JSON\_QRCodeEventCond

JSON message about the condition of actively getting QR code scanning events

```

{
    "QRCodeEventCond": {
        "searchID": "",
/*required, string, search ID, which is used to check whether the current
search requester is the same as the previous one. If they are the same, the
search record will be stored in the device to speed up the next search*/
        "searchResultPosition": 0,
/*required, int, the start position of search result in the result list. In a
single search, if you cannot get all the records in the result list, you can
mark the end position and get the following records after the marked position
in the next search. If the maximum number of totalMatches supported by the
device is M and the number of totalMatches stored in the device now is N
(N<=M), the valid range of this node is 0 to N-1*/
        "maxResults": 30,
/*required, int, the maximum number of search results that can be obtained by
calling the URI this time. If the value of maxResults is greater than that
defined in the device capability, the value in the capability will be returned.
In this case, the device will not return error*/
        "startTime": "2016-12-12T17:30:08+08:00",
/*optional, string, start time (UTC time)*/
        "endTime": "2017-12-12T17:30:08+08:00",
/*optional, string, end time (UTC time)*/
        "picEnable": true,
/*optional, boolean, whether to upload the picture along with the event
information: true (all matched events will be uploaded with pictures if there
are any), false (all matched events will be uploaded without pictures). If this
node is not configured, the default value is true*/
        "beginSerialNo": 1,
/*optional, int, start serial No.*/
        "endSerialNo": 1
/*optional, int, end serial No.*/
    }
}

```

### F.144 JSON\_RegionCalibrationCfg

JSON message about the calibration parameters of the temperature measurement area

```
{  
    "enabled":true,  
    /*required, boolean, whether to enable calibration: true, false*/  
    "FaceFrameCoordinate":{  
        /*optional, object, coordinate of the face frame, the value is normalized to a  
        number between 0 and 1000*/  
        "height":1,  
        /*optional, int, height, value range: [0,1000]*/  
        "width":2,  
        /*optional, int, width, value range: [0,1000]*/  
        "x":5,  
        /*optional, int, X-coordinate, value range: [0,1000]*/  
        "y":10  
        /*optional, int, Y-coordinate, value range: [0,1000]*/  
    }  
}
```

### F.145 JSON\_RegionCoordinate

JSON message about the parameters of the temperature measurement area

```
{  
    "RegionCoordinate":[]  
    /*required, array of object, coordinates of vertexes of the polygon. The number  
    of vertexes of the polygon is between 3 and 10*/  
    [{"x":1,  
     /*optional, int, X-coordinate, the value is between 0 and 1000*/  
     "y":2  
     /*optional, int, Y-coordinate, the value is between 0 and 1000*/  
    }]  
}
```

### F.146 JSON\_RemoteCheck

Message about the parameters of verifying the access control event remotely in JSON format.

```
{  
    "RemoteCheck":{  
        "serialNo": ,  
        /*required, int, event serial No. which should be the same as that in the event  
        information message for uploading*/  
        "checkResult":""  
        /*required, string, verification result: "success"-verified, "failed"-
```

```
verification failed*/
    "info": ""
/*optional, string, additional information*/
}
}
```

### F.147 JSON\_RemoteControlBuzzer

RemoteControlBuzzer message in JSON format

```
{
  "RemoteControlBuzzer": {
    "cmd": ""
/*required, string, command: "start"-start buzzing, "stop"-stop buzzing*/
  }
}
```

### F.148 JSON\_RemoteControllerModeCfg

JSON message about the parameters of the keyfob control mode.

```
{
  "RemoteControllerModeCfg": {
    "mode": ""
/*required, string, keyfob control mode: "oneToOne"-one-to-one mode (default,
the keyfob can only control one device), "oneToMany"-one-to-many mode (the
keyfob can control multiple devices)*/
  }
}
```

### F.149 JSON\_RemoteControllerModeCfgCap

JSON message about the configuration capability of the keyfob control mode

```
{
  "RemoteControllerModeCfgCap": {
    "mode": {
/*required, keyfob control mode: "oneToOne"-one-to-one mode (default, the
keyfob can only control one device), "oneToMany"-one-to-many mode (the keyfob
can control multiple devices)*/
      "@opt": ["oneToOne", "oneToMany"]
    }
  }
}
```

## F.150 JSON\_ResponseStatus

JSON message about response status

```
{
  "requestURL":"",
  /*optional, string, request URL*/
  "statusCode": ,
  /*optional, int, status code*/
  "statusString":"",
  /*optional, string, status description*/
  "subStatusCode":"",
  /*optional, string, sub status code*/
  "errorCode": ,
  /*required, int, error code, which corresponds to subStatusCode, this field is required when statusCode is not 1. The returned value is the transformed decimal number*/
  "errorMsg":"",
  /*required, string, error details, this field is required when statusCode is not 1*/
  "MErrCode": "0xFFFFFFFF",
  /*optional, string, error code categorized by functional modules*/
  "MErrDevSelfEx": "0xFFFFFFFF"
  /*optional, string, extension of MErrCode. It is used to define the custom error code, which is categorized by functional modules*/
}
```

## F.151 JSON\_RFCardCfg

RFCardCfg message in JSON format

```
{
  "RFCardCfg":[{
    "cardType":"",
    /*required, string, card type: "EMCard"-EM card, "M1Card"-M1 card, "CPUCard"-CPU card, "IDCard"-ID card, "DesfireCard"-DESFire card, "FelicaCard"-Felica card*/
    "enabled":
    /*required, boolean, whether to enable RF card recognition: "true"-yes, "false"-no*/
  }]
}
```

## F.152 JSON\_RFCardCfgCap

RFCardCfgCap capability message in JSON format

```
{
    "RFCardCfgCap": {
        "cardType": {
            /*required, string, card type: "EMCard"-EM card, "M1Card"-M1 card, "CPUCard"-CPU card, "IDCard"-ID card, "DesfireCard"-DESFire card, "FelicaCard"-Felica card*/
            "@opt": ["EMCard", "M1Card", "CPUCard", "IDCard"]
        },
        "enabled": {
            /*required, boolean, whether to enable RF card recognition: "true"-yes, "false"-no*/
            "@opt": [true, false]
        }
    }
}
```

## F.153 JSON\_RuleInfo

RuleInfo message in JSON format

```
{
    "RuleInfo": {
        "reqAdminRights": ,
        /*required, boolean, whether the administrator permission is required: "true"-yes, "false"-no*/
        "enableCardNoLenAuto": ,
        /*optional, boolean, whether to enable length self-adaption of the card serial No. The priority of this field is higher than len*/
        "RuleList": [
            /*rule list, which contains rules for collecting different types of data*/
            {
                "dataType": "",
                /*required, string, data type: "name", "employeeNo"-employee No., "IDCardNo"-ID card No., "IDCardSerialNo"-ID card serial No., "IDCardDetails"-ID card details, "card", "fingerprint"-fingerprint, "face"*/
                "enable": ,
                /*required, boolean, whether to collect and display: "true"-collect and display, "false"-not collect and display*/
                "uniqueCheck": ,
                /*dependency, boolean, whether to enable uniqueness verification: "true"-yes, "false" (default) or this field is not returned-no. This field is valid when dataType is "name". For other data types, this field is the read-only optional parameter*/
                "len": ,
                /*dependency, integer, data length, this field is valid when dataType is "name", "employeeNo" or "card". The default data length of name is 128. For other data types, this field is the read-only optional parameter. If it is not supported, this field will not be returned*/
                "num": ,
                /*dependency, integer, number of collected data, this field is valid when dataType is "fingerprint" or "card"*/
            }
        ]
    }
}
```

```

        "fingerprintIDs":  

/*dependency, integer, ID list of fingerprints that need to be collected, this  

field is valid when dataType is "fingerprint"*/  

    }],  

    "enableLocalIssueCard": true,  

/*optional, boolean, whether to enable issuing smart cards locally*/  

    "isLocalStorage": false  

/*optional, boolean, whether to store face picture and fingerprint information  

in the device locally*/  

}  

}

```

### **F.154 JSON\_SafetyHelmetDetection**

JSON message about parameters of hard hat detection

```

{
  "SafetyHelmetDetection": {
    "enable": true,  

/*optional, boolean, whether to enable hard hat detection: true-yes, false-no  

(default)*/
    "noHelmetStrategy": ""  

/*optional, string, door control strategy when not wearing hard hat is  

detected: "normal"-open the door to allow access, "forbidden"-access is  

prohibited*/
  }
}

```

### **F.155 JSON\_SafetyHelmetDetectionCap**

JSON message about the configuration capability of hard hat detection

```

{
  "SafetyHelmetDetectionCap": {
    "enable": {
      /*optional, boolean, whether to enable hard hat detection: true-yes, false-no  

(default)*/
      "@opt": [true, false]
    },
    "noHelmetStrategy": {
      /*optional, string, door control strategy when not wearing hard hat is  

detected: "normal"-open the door to allow access, "forbidden"-access is  

prohibited*/
      "@opt": ["normal", "forbidden"]
    }
  }
}

```

## F.156 JSON\_SearchFaceRecordCond

Message about conditions of searching for face records, and it is in JSON format.

```
{
    "searchResultPosition": "",  

    /*required, initial position of search result list, integer32 type. When there  

    are multiple records, and cannot get all records in one time searching, you can  

    search the records followed specified position for next search. For video  

    intercom devices, this field can only be set to 0 as the picture will be  

    returned along with the message*/  

    "maxResults": "",  

    /*required, int32 type, maximum number of records for single searching. If  

maxResults exceeds the range defined by the device capability, the device will  

    return the maximum number of records according to the device capability and  

    will not return error. For video intercom devices, this field can only be set  

    to 1 as the picture will be returned along with the message*/  

    "faceLibType": "",  

    /*required, face picture library type: "blackFD"-list library, "staticFD"-  

    static library, string type, the maximum size is 32 bytes*/  

    "FDID": "",  

    /*required, face picture library ID, string type, the maximum size is 63*/  

    "FPID": "",  

    /*optional, string type, face record ID, it can be generated by device or  

    inputted. If it is inputted, it should be the unique ID with the combination of  

    letters and digits, and the maximum length is 63 bytes; if it is generated by  

    the device automatically, it is the same as the employee No. (person ID)*/  

    "startTime": "",  

    /*optional, start time, ISO8601 time format, string type, the maximum size is  

    32 bytes*/  

    "endTime": "",  

    /*optional, end time, ISO8601 time format, string type, the maximum size is 32  

    bytes*/  

    "name": "",  

    /*optional, name, string type, the maximum size is 96 bytes*/  

    "gender": "",  

    /*optional, gender: male, female, unknown, string type, the maximum size is 10*/  

    "city": "",  

    /*optional, city code of birth for the person in the face picture, string type,  

    the maximum size is 32 bytes*/  

    "certificateType": "",  

    /*optional, string type, the maximum size is 10 bytes, certificate type:  

    "officerID"-officer ID, "ID"-identify card, passport, other*/  

    "certificateNumber": ""  

    /*optional, certificate No., string, the maximum size is 32 bytes*/  

    "isInLibrary": "yes",  

    /*optional, string type, whether the picture is in library (whether modeling is  

    successful): unknown, no, yes*/  

    "isDisplayCaptureNum": true,  

    /*optional, boolean type, whether to display number of captured pictures, true:  

    display, false: hide, by default it is false*/
}
```

```

    "rowKey":"",
/*optional, string type, face picture library main key. Search by rowKey can be
more efficient, the maximum size is 64 bytes*/
    "transfer":true
/*optional, boolean type, whether to enable transfer*/
}

```

### F.157 JSON\_SearchFaceRecordResult

Message about result of searching for face record.

```

{
    "requestURL": "",
    "statusCode": ,
    "statusString": "",
    "subStatusCode": "",
    "errorCode": ,
    "errorMsg": "",

/*see the description of this node and above nodes in the message of
JSON_ResponseStatus*/
    "responseStatusStrg": "",

/*optional, searching status: "OK"-searching ended, "NO MATCHES"-no data found,
"MORE"-searching, string type, the max. size is 32 bytes. It is valid only when
errorCode is 1 and errorMsgis ok*/
    "searchResultPosition": "",

/*optional, initial position of search result list, integer32 type. It is valid
only when errorCode is 1 and errorMsgis ok*/
    "numOfMatches": , 

/*optional, returned number of results for current search, integer32. It is
valid only when errorCode is 1 and errorMsgis ok*/
    "totalMatches": , 

/*optional, total number of matched results, integer32. It is valid only when
errorCode is 1 and errorMsgis ok*/
    "MatchList": [
/*optional, searched matched data information, array. It is valid only when
errorCode is 1 and errorMsgis ok*/
        {
            "FPIID":"",
/*optional, string type, face record ID (it is the same as the employee No.
(person ID)), the maximum length is 63 bytes*/
                "FDID":"test",
/*optional, string, face picture library ID, read-only*/
                "FDName":"List Library A",
/*optional, string, face picture library name, read-only*/
                "faceURL":"",
/*optional, face picture URL, string type, the maximum size is 128 bytes*/
                "name":"",
/*required, name of person in the face picture, string type, the maximum size
is 96 bytes*/
                "gender": ""
        }
    ]
}

```

```
/*optional, gender of person in the face picture: male, female, unknown, string
type, the maximum size is 32 bytes*/
    "bornTime": "",

/*required, birthday of person in the face picture, ISO8601 time format, string
type, the maximum size is 20 bytes*/
    "city": "",

/*optional, city code of birth for the person in the face picture, string type,
the maximum size is 32 bytes*/
    "certificateType": "",

/*optional, string type, the max. size is 10 bytes, certificate type:
"officerID"-officer ID, "ID"-identify card, passport, other*/
    "certificateNumber": "",

/*optional, certificate No., string, the max. size is 32 bytes*/
    "caseInfo": "",

/*optional, case information, string type, the max. size is 192 bytes, it is
valid when faceLibType is blackFD.*/
    "tag": "",

/*optional, custom tag, up to 4 tags, which are separated by commas, string
type, the max. size is 195 bytes, it is valid when faceLibType is blackFD.*/
    "address": "",

/*optional, person address, string type, the max. size is 192 bytes, it is
valid when faceLibType is staticFD.*/
    "customInfo": "",

/*optional, custom information, string type, the max. size is 192 bytes, it is
valid when faceLibType is staticFD.*/
    "modelData":""

/*optional, string type, target model data, non-modeled binary data needs to be
encrypted by base64 during transmission*/
    "isInLibrary": "yes",

/*optional, string type, whether the picture is in library (whether modeling is
successful): unknown, no, yes*/
    "captureNum": 12,

/*optional, int, number of captured pictures*/
    "rowKey": "",

/*optional, string type, face picture library main key. Search by rowKey can be
more efficient, the maximum size is 64 bytes*/
    "saveFacePic": true

/*optional, boolean, whether to save face pictures*/
    }

]
}
```

### See Also

[JSONResponseStatus](#)

## F.158 JSON\_SearchTaskCond

SearchTaskCond message in JSON format

```
{
    "SearchTaskCond": {
        "searchID": "",
        /*required, string, search ID which is used to check whether the upper-layer
        clients are the same one*/
        "searchResultPosition": ,
        /*required, integer32, the start position of the search result in the result
        list. When there are multiple records and you cannot get all search results at
        a time, you can search for the records after the specified position next time.
        If the device returns the picture along with the response message, this field
        should be between 0 and totalMatches*/
        "maxResults": ,
        /*required, integer32, the maximum number of results that can be obtained by
        calling the URL at a time. If the device returns the picture along with the
        response message, this field can only be set to 1*/
        "captureNoList": ,
        /*optional, integer, collection No. list. If the collection No. is not
        configured, it will search all data according to searchResultPosition*/
        "searchType": ""
        /*optional, search type: "new"-search and only return newly added data,
        "modified"-search and only return edited data. By default all data will be
        searched*/
    }
}
```

### F.159 JSON\_SearchTaskResponse

SearchTaskResponse message in JSON format

```
{
    "SearchTaskResponse": {
        "searchID": "",
        /*required, string, search ID which is used to check whether the upper-layer
        clients are the same one*/
        "responseStatusStrg": "",
        /*optional, string, searching status: "OK"-searching completed, "NO MATCH"-no
        matched results, "MORE"-searching for more results*/
        "numOfMatches": ,
        /*optional, integer32, number of returned results this time*/
        "totalMatches": ,
        /*optional, integer32, total number of matched results*/
        "DataCollections": [
            /*optional, array, searched matched data information*/
            "lastCaptureNo": ,
            /*required, integer, last collection No., it is used to check whether there is
            data lost*/
            "captureNo": ,
            /*required, integer, current collection No.*/
            "name": "",
            /*optional, string, name*/
        ]
    }
}
```

```
        "employeeNo":"",
/*optional, string, employee No.*/
        "IDCardNo":"",
/*optional, string, ID card No.*/
        "CardNoList":[{
/*optional, string, card No. list*/
            "cardNo":"",
            "cardType": "TypeA_M1"
/*optional, string, card type: "TypeA_M1", "TypeA_CPU", "TypeB", "ID_125K",
"FelicaCard", "DesfireCard"*/
        }],
        "FingerprintList":[{
            "fingerprintID": ,
            /*optional, integer, fingerprint No.*/
            "fingerprint": ""
/*optional, string, fingerprint information which is encoded using base64*/
        }],
        "FaceFeature":{
/*optional, feature information of face picture matting*/
            "Region":{
/*required, area coordinates of face picture matting, it is a rectangle*/
                "height": ,
/*required, float, height*/
                "width": ,
/*required, float, width*/
                "x": ,
/*required, float, X-coordinate of the left corner*/
                "y": 
/*required, float, Y-coordinate of the left corner*/
            },
            "LeftEyePoint":{
/*optional, coordinates of the left eye*/
                "x": ,
/*required, float, X-coordinate, it is between 0.000 and 1*/
                "y": 
/*required, float, Y-coordinate, it is between 0.000 and 1*/
            },
            "RightEyePoint":{
/*optional, coordinates of the right eye*/
                "x": ,
/*required, float, X-coordinate, it is between 0.000 and 1*/
                "y": 
/*required, float, Y-coordinate, it is between 0.000 and 1*/
            },
            "LeftMouthPoint":{
/*optional, coordinates of the left mouth corner*/
                "x": ,
/*required, float, X-coordinate, it is between 0.000 and 1*/
                "y": 
/*required, float, Y-coordinate, it is between 0.000 and 1*/
            },
            "RightMouthPoint":{
```

```
/*optional, coordinates of the right mouth corner*/
    "x": ,
/*required, float, X-coordinate, it is between 0.000 and 1*/
    "y": ,
/*required, float, Y-coordinate, it is between 0.000 and 1*/
},
    "NoseTipPoint":{
/*optional, coordinates of the nose*/
    "x": ,
/*required, float, X-coordinate, it is between 0.000 and 1*/
    "y": ,
/*required, float, Y-coordinate, it is between 0.000 and 1*/
}
},
    "riskDataMark": ,
/*optional, boolean, whether to mark risk data: "true"-mark the data as the
risk data and person and ID comparison failed, "false" or this field is not
returned-the data is normal*/
    "dataType":"",
/*optional, string, data type and status: "new"-newly added data, "modified"-_
edited data, "normal"-unchanged data*/
    "IdentityInfo":{
/*identity information*/
    "chnName":"",
/*optional, string, Chinese name*/
    "enName":"",
/*optional, string, English name*/
    "sex":"",
/*optional, string, gender: "male", "female"*/
    "birth":"",
/*optional, string, data of birth, e.g., "1990-02-24"*/
    "addr":"",
/*optional, string, address*/
    "IDCardNo":"",
/*optional, string, ID card No.*/
    "issuingAuthority":"",
/*optional, string, issuing authority*/
    "startDate":"",
/*optional, string, start date of validity period*/
    "endDate":"",
/*optional, string, end date of validity period*/
    "passNo":"",
/*optional, string, entry-exit permit No.*/
    "issueNumber":"",
/*optional, string, issued times*/
    "certificateType":"",
/*optional, string, certificate type*/
    "permanentResidenceCardNo":"",
/*optional, string, permanent resident visa No.*/
    "nationalityOrAreaCode":"",
/*optional, string, country/region code*/
    "version":""
```

```

/*optional, string, certificate version No.*/
    "receivingAuthorityCode":"",
/*optional, string, acceptance authority code*/
    "FingerprintList":[{
        "fingerprint":""
    /*optional, string, fingerprint information, which should be encoded by Base64*/
    }],
    "pic":""
/*optional, string, certificate picture information, which should be encoded by
Base64, encrypted and decrypted by a specific decryption library*/
},
    "CardIssueStatus":[{
/*optional, issuing status list of cards containing face pictures and
fingerprints*/
        "cardNo":"",
/*optional, string, card information*/
        "face":true,
/*optional, boolean, card issuing status of the face picture: true-with card
issued, false-without card issued*/
        "fingerprint1":true,
/*optional, boolean, card issuing status of the fingerprint 1: true-with card
issued, false-without card issued*/
        "fingerprint2":true
/*optional, boolean, card issuing status of the fingerprint 2: true-with card
issued, false-without card issued*/
    }]
},
}
}

```

## F.160 JSON\_SectionEncryption

JSON message about section encryption parameters

```

{
    "SectionEncryption": {
        "sectionNo": ,
/*required, integer, section No.*/
        "keyType": "",
/*required, string, key types: "private"-private key, "normal"-other valid
keys*/
        "password": ""
/*depend, string, a hexadecimal verification key, this field is valid only when
the keyType is "normal"*/
        "newKeyType": "",
/*required, string, new key types: "private"-private key, "normal"-other valid
keys*/
        "KeyA": "",
/*depend, string, a hexadecimal password of key A, this field is valid only
when the keyType is "normal"*/
    }
}

```

```

    "KeyB": "",  

    /*depend, string, a hexadecimal password of key B, this field is valid only  

    when the keyType is "normal"*/  

    "controlBits":  

    /*depend, a hexadecimal control bit, this field is valid only when the keyType  

    is "normal"*/  

    }  

}
}

```

### F.161 JSON\_SetFaceRecord

Message about the condition of setting the face record, and it is in JSON format.

```

{
    "faceURL": "",  

    /*optional, string type, picture storage URL inputted when uploading the face  

    picture by URL, the maximum length is 256 bytes*/  

    "faceLibType": "",  

    /*required, string type, face picture library type: "blackFD"-list library,  

    "staticFD"-static library, the maximum length is 32 bytes*/  

    "FDID": "",  

    /*required, string type, face picture library ID, the maximum length is 63  

    bytes*/  

    "FPID": "",  

    /*optional, string type, face record ID, it can be generated by the device or  

    inputted. If it is inputted, it should be the unique ID with the combination of  

    letters and digits, and the maximum length is 63 bytes; if it is generated by  

    the device automatically, it is the same as the employee No. (person ID)*/  

    "deleteFP": ,  

    /*optional, boolean type, whether to delete the face record: "true"-yes. This  

    node is required when the face record needs to be deleted; for adding or  

    editing the face record, this node should be set to NULL*/  

    "name": "",  

    /*required, string type, name of the person in the face picture, the maximum  

    length is 96 bytes*/  

    "gender": "",  

    /*optional, string type, gender of the person in the face picture: "male",  

    "female", "unknown", the maximum length is 32 bytes*/  

    "bornTime": "",  

    /*required, string type, date of birth of the person in the face picture in  

    ISO8601 time format, the maximum length is 20 bytes*/  

    "city": "",  

    /*optional, string type, code of the city of birth for the person in the face  

    picture, the maximum length is 32 bytes*/  

    "certificateType": "",  

    /*optional, string type, ID type: "officerID"-officer ID, "ID"-ID card. The  

    maximum length is 10 bytes*/  

    "certificateNumber": "",  

    /*optional, string type, ID No., the maximum length is 32 bytes*/  

    "caseInfo": "",  

}
}

```

```

/*optional, string type, case information, the maximum length is 192 bytes, it
is valid when faceLibType is "blackFD"*/
    "tag":"",
/*optional, string type, custom tag, up to 4 tags can be added and they should
be separated by commas, the maximum length of each tag is 48 bytes, and the
maximum length of this node is 195 bytes. It is valid when faceLibType is
"blackFD"*/
    "address":"",
/*optional, string type, person address, the maximum length is 192 bytes, it is
valid when faceLibType is "staticFD"*/
    "customInfo":"",
/*optional, string type, custom information, the maximum length is 192 bytes,
it is valid when faceLibType is "staticFD"*/
    "modelData":"",
/*optional, string type, target model data, non-modeled binary data needs to be
encrypted by base64 during transmission*/
    "PicFeaturePoints":{},
/*optional, array of object, feature points to be applied. If the device only
supports three types of feature points, when the platform applies more than
three types of feature points, the device will not return error information*/
        "featurePointType":"face",
/*required, string, feature point type: "face", "leftEye" (left eye),
"rightEye" (right eye), "leftMouthCorner" (left corner of mouth),
"rightMouthCorner" (right corner of mouth), "nose"*/
        "coordinatePoint":{
/*required, object, coordinates of the feature point*/
            "x":1,
/*required, int, normalized X-coordinate which is between 0 and 1000*/
            "y":1,
/*required, int, normalized Y-coordinate which is between 0 and 1000*/
            "width":1,
/*required, int, width which is between 0 and 1000. This node is required when
featurePointType is "face"*/
            "height":1
/*required, int, height which is between 0 and 1000. This node is required when
featurePointType is "face"*/
        }
    },
    "saveFacePic": true
/*optional, boolean, whether to save face pictures*/
}

```

### F.162 JSON\_SingleFPLibInfo

Message about the information of a face picture library, and it is in JSON format.

```
{
    "requestURL": "",
    "statusCode": "",
    "statusString": "",
```

```
"subStatusCode": "",  
"errorCode": "",  
"errorMsg": "",  
/*see the description of this node and above nodes in the message of  
JSON_ResponseStatus*/  
"faceLibType": "",  
/*optional, face picture library type: "blackFD"-list library, "staticFD"-  
static library, string type, the max. string length is 32 bytes*/  
"name": "",  
/*optional, face picture library name, string type, the max. string length is  
48 bytes*/  
"customInfo": "",  
/*optional, custom information, string type, the max. string length is 192  
bytes*/  
"libArmingType": "armingLib",  
/*optional, string, arming type of the list library: "armingLib" (armed face  
picture library), "nonArmingLib" (not armed face picture library). The default  
value is "armingLib"*/  
"libAttribute": "blackList",  
/*optional, string, library type: "blackList" (blocklist library), "VIP" (VIP  
library), "passerby" (passerby library). The passerby library cannot be  
deleted*/  
"personnelFileEnabled": true  
/*optional, boolean, whether to enable personnel archive configuration, read-  
only*/  
}
```

### See Also

[JSON\\_ResponseStatus](#)

## F.163 JSON\_TemperatureMeasurementCfg

JSON message about the temperature measurement parameters

```
{  
    "showTemperatureInfo":true,  
    /*optional, boolean, whether to display the temperature information: true,  
    false*/  
    "saveThermalPicture":true,  
    /*optional, boolean, whether to save the thermal picture: true, false*/  
    "uploadThermalPicture":true,  
    /*optional, boolean, whether to upload the thermal picture: true, false*/  
    "lowTemperatureEnabled":true  
    /*optional, boolean, whether to enable temperature measurement in the low-  
    temperature environment: true, false*/  
}
```

## F.164 JSON\_UploadFailedDetails

JSON message about the details of failing to upload the user list of offline collection

```
{
  "UploadFailedDetails": {
    "description": ""
  }
}
```

## F.165 JSON\_UserInfo

JSON message about the person information

```
{
  "UserInfo": {
    "employeeNo": "",
    /*required, string, employee No. (person ID)*/
    "deleteUser": ,
    /*optional, boolean, whether to delete the person: "true"-yes. This node is required only when the person needs to be deleted; for adding or editing person information, this node can be set to NULL*/
    "name": "",
    /*optional, string, person name*/
    "userType": "",
    /*required, string, person type: "normal"-normal person (household), "visitor", "blackList"-person in blocklist*/
    "closeDelayEnabled": ,
    /*optional, boolean, whether to enable door close delay: "true"-yes, "false"-no*/
    "Valid": {
      /*required, parameters of the effective period, the effective period can be a period of time between 1970-01-01 00:00:00 and 2037-12-31 23:59:59*/
      "enable": ,
      /*required, boolean, whether to enable the effective period: "false"-disable, "true"-enable. If this node is set to "false", the effective period is permanent*/
      "beginTime": "",
      /*required, start time of the effective period (if timeType does not exist or is "local", the beginTime is the device local time, e.g., 2017-08-01T17:30:08; if timeType is "UTC", the beginTime is UTC time, e.g., 2017-08-01T17:30:08+08:00)*/
      "endTime": "",
      /*required, end time of the effective period (if timeType does not exist or is "local", the endTime is the device local time, e.g., 2017-08-01T17:30:08; if timeType is "UTC", the endTime is UTC time, e.g., 2017-08-01T17:30:08+08:00)*/
      "timeType": ""
    }
  }
}
```

```

/*optional, string, time type: "local"- device local time, "UTC"- UTC time*/
    },
    "belongGroup":"",
/*optional, string, group*/
    "password":"",
/*optional, string, password*/
    "doorRight":"",
/*optional, string, No. of the door or lock that has access permission, e.g.,
"1,3" indicates having permission to access door (lock) No. 1 and No. 3*/
    "RightPlan": [{

/*optional, door permission schedule (lock permission schedule)*/
        "doorNo": ,
/*optional, integer, door No. (lock ID)*/
        "planTemplateNo":""
/*optional, string, schedule template No.*/
    }],
    "maxOpenDoorTime": ,
/*optional, integer, maximum authentication attempts, 0-unlimited*/
    "openDoorTime": ,
/*optional, integer, read-only, authenticated attempts*/
    "roomNumber": ,
/*optional, integer, room No.*/
    "floorNumber": ,
/*optional, integer, floor No.*/
    "doubleLockRight": ,
/*optional, boolean, whether to have the permission to open the double-locked
door: "true"-yes, "false"-no*/
    "localUIRight": ,
/*optional, boolean, whether to have the permission to access the device local
UI: "true"-yes, "false"-no*/
    "localUIUserType":"",
/*optional, string, user type of device local UI: "admin" (administrator),
"operator", "viewer" (normal user). This node is used to distinguish different
users with different operation permissions of device local UI*/
    "userVerifyMode":"",
/*optional, string, person authentication mode: "cardAndPw"-card+password,
"card"-card, "cardOrPw"-card or password, "fp"-fingerprint, "fpAndPw"-fingerprint+password,
"fpOrCard"-fingerprint or card, "fpAndCard"-fingerprint+card, "fpAndCardAndPw"-fingerprint+card+password,
"faceOrFpOrCardOrPw"-face or fingerprint or card or password, "faceAndFp"-face+fingerprint, "faceAndPw"-face+password,
"faceAndCard"-face+card, "face"-face, "employeeNoAndPw"-employee No.+password,
"fpOrPw"-fingerprint or password, "employeeNoAndFp"-employee No.+fingerprint,
"employeeNoAndFpAndPw"-employee No.+fingerprint+password,
"faceAndFpAndCard"-face+fingerprint+card, "faceAndPwAndFp"-face+password+fingerprint,
"employeeNoAndFace"-employee No.+face, "faceOrfaceAndCard"-face or face+card,
"fpOrface"-fingerprint or face, "cardOrfaceOrPw"-card or face or password,
"cardOrFace"-card or face, "cardOrFaceOrFp"-card or face or fingerprint,
"cardOrFpOrPw"-card or fingerprint or password. The priority of
the person authentication mode is higher than that of the card reader
authentication mode*/
    "checkUser": ,
/*optional, boolean, whether to verify the duplicated person information:

```

```

"false"-no, "true"-yes. If checkUser is not configured, the device will verify
the duplicated person information by default. When there is no person
information, you can set checkUser to "false" to speed up data applying;
otherwise, it is not recommended to configure this node*/
    "addUser": ,
/*optional, boolean type, whether to add the person if the person information
being edited does not exist: "false"-no (if the device has checked that the
person information being edited does not exist, the failure response message
will be returned along with the error code), "true"-yes (if the device has
checked that the person information being edited does not exist, the success
response message will be returned, and the person will be added). If this node
is not configured, the person will not be added by default*/
    "dynamicCode": "123456",
/*optional, string, dynamic permission code, this node is write-only*/
    "callNumbers": ["","",""],
/*optional, string type, room No. list to be called, by default, its format is
X-X-X-X (e.g., 1-1-1-401), which is extended from roomNumber; for standard SIP,
it can be the SIP number*/
    "floorNumbers": [ , ],
/*optional, integer type, floor No. list, which is extended from floorNumber*/
    "numOfFace": ,
/*optional, read-only, number of linked face pictures. If this field is not
returned, it indicates that this function is not supported*/
    "numOfFP": ,
/*optional, read-only, number of linked fingerprints. If this field is not
returned, it indicates that this function is not supported*/
    "numOfCard": ,
/*optional, read-only, number of linked cards. If this field is not returned,
it indicates that this function is not supported*/
    "gender":"",
/*optional, string, gender of the person in the face picture: "male", "female",
"unknown"*/
    "PersonInfoExtends":{},
/*optional, array of object, extended fields for the additional person
information. This node is used to configure the extended person information
displayed on the device's UI. For MinMoe series facial recognition terminals,
currently only one value node can be supported for displaying the employee No.
and the node id is not supported*/
        "id":1,
/*optional, int, extended ID of the additional person information, value range:
[1,32]. It corresponds to the id in the message of the request URI /ISAPI/
AccessControl/personInfoExtendName?format=json and is used to link the value of
the node value and its name (the node name in the message of the request URI /
ISAPI/AccessControl/personInfoExtendName?format=json). If the node id does not
exist, the ID will start from 1 by default according to the array order*/
        "value":"",
/*optional, string, extended content of the additional person information*/
    },
    "operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal*/
    "terminalNoList": [1],
/*optional, array, terminal ID list, this node is required when operation type

```

```
is "byTerminal"; currently, only one terminal is supported*/
    "groupId":1,
/*optional, int, department No. of local time and attendance*/
    "localAtndPlanTemplateId":1
/*optional, int, schedule template of local time and attendance. If this node
exist, it indicates that there are shift schedule settings by individual. If id
is 0, it indicates canceling the shift schedule of the person*/
    }
}
```

### F.166 JSON\_UserInfoCount

UserInfoCount message in JSON format

```
{
  "UserInfoCount":{
    "userNumber":  
/*required, integer, number of persons*/
  }
}
```

### F.167 JSON\_UserInfoDelCond

JSON message about user information to be deleted

```
{
  "UserInfoDelCond":{
    "EmployeeNoList":[]  
/*optional, person ID list (if this node does not exist or is set to NULL, it
indicates deleting all person information)*/
    "employeeNo":""  
/*optional, string, employee No. (person ID)*/
    ],
    "operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal*/
    "terminalNoList": [1]
/*optional, array, terminal ID list, this node is required when operation type
is "byTerminal"; currently, only one terminal is supported*/
  }
}
```

### F.168 JSON\_UserInfoDetail

JSON message about user information

```
{
  "UserInfoDetail":{
```

```
"mode":"",
/*required, string, deleting mode: "all"-delete all, "byEmployeeNo"-delete by
employee No. (person ID)*/
"EmployeeNoList":{},
/*optional, person ID list, if this node does not exist or is null, it
indicates deleting all person information (including linked cards and
fingerprints) and permissions*/
"employeeNo":"",
/*optional, string, employee No. (person ID), it is valid when mode is
"byEmployeeNo"*/
},
"operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal, "byOrg"-by
organization, "byTerminalOrg"-by terminal organization*/
"terminalNoList": [ 1, 2, 3, 4 ],
/*optional, array, terminal ID list, this node is required when operation type
is "byTerminal" or "byTerminalOrg"*/
"orgNoList": [ 1, 2, 3, 4 ]
/*optional, array, organization ID list, this node is required when operation
type is "byOrg" or "byTerminalOrg"*/
}
}
```

### F.169 JSON\_UserInfoDetailDeleteProcess

UserInfoDetailDeleteProcess message in JSON format

```
{
  "UserInfoDetailDeleteProcess": {
    "status": ""
  }
}
```

### F.170 JSON\_UserInfoSearch

UserInfoSearch message in JSON format

```
{
  "UserInfoSearch": {
    "searchID": "",
    /*required, string type, search ID, which is used to confirm the upper-level
    platform or system. If the platform or the system is the same one during two
    searching, the search history will be saved in the memory to speed up next
    searching*/
    "responseStatusStrg": "",
    /*required, string, search status: "OK"-searching completed, "NO MATCH"-no
    matched results, "MORE"-searching for more results*/
    "numOfMatches": ,
  }
}
```

```

/*required, integer32, number of returned results this time*/
    "totalMatches": ,
/*required, integer32, total number of matched results*/
    "UserInfo": [
/*optional, person information*/
        "employeeNo":"",
/*required, string, employee No. (person ID)*/
        "name":"",
/*optional, string, person name*/
        "userType":"",
/*required, string, person type: "normal"-normal person (household), "visitor",
"blackList"-person in blocklist*/
        "closeDelayEnabled": ,
/*optional, boolean, whether to enable door close delay: "true"-yes, "false"-no*/
        "Valid": {
/*required, parameters of the effective period*/
            "enable":"",
/*required, boolean, whether to enable the effective period: "false"-disable,
"true"-enable. If this node is set to "false", the effective period is
permanent*/
            "beginTime":"",
/*required, start time of the effective period (if timeType does not exist or
is "local", the beginTime is the device local time, e.g., 2017-08-01T17:30:08;
if timeType is "UTC", the beginTime is UTC time, e.g.,
2017-08-01T17:30:08+08:00)*/
            "endTime":"",
/*required, end time of the effective period (if timeType does not exist or is
"local", the endTime is the device local time, e.g., 2017-08-01T17:30:08; if
timeType is "UTC", the endTime is UTC time, e.g., 2017-08-01T17:30:08+08:00)*/
            "timeType":""
/*optional, string, time type: "local"- device local time, "UTC"- UTC time*/
        },
        "belongGroup":"",
/*optional, string, group*/
        "password":"",
/*optional, string, password*/
        "doorRight":"",
/*optional, string, No. of door or lock that has access permission, e.g., "1,3"
indicates having permission to access door (lock) No. 1 and No. 3*/
        "RightPlan":[{
/*optional, access permission schedule of the door or lock*/
            "doorNo": ,
/*optional, integer, door No. (lock ID)*/
            "planTemplateNo":""
/*optional, string, schedule template No.*/
        }],
        "maxOpenDoorTime": ,
/*optional, integer, the maximum authentication attempts, 0-unlimited*/
        "openDoorTime": ,
/*optional, integer, read-only, authenticated attempts*/
        "roomNumber": ,

```

```

/*optional, integer, room No.*/
    "floorNumber": ,
/*optional, integer, floor No.*/
    "doubleLockRight": ,
/*optional, boolean, whether to have the permission to open the double-locked
door: "true"-yes, "false"-no*/
    "localUIRight": ,
/*optional, boolean, whether to have the permission to access the device local
UI: "true"-yes, "false"-no*/
    "localUIUserType":"",
/*optional, string, user type of device local UI: "admin" (administrator),
"operator", "viewer" (normal user). This node is used to distinguish different
users with different operation permissions of device local UI*/
    "userVerifyMode":"",
/*optional, string, person authentication mode: "cardAndPw"-card+password,
"card"-card, "cardOrPw"-card or password, "fp"-fingerprint, "fpAndPw"-fingerprint+password,
"fpOrCard"-fingerprint or card, "fpAndCard"-fingerprint+card, "fpAndCardAndPw"-fingerprint+card+password,
"faceOrFpOrCardOrPw"-face or fingerprint or card or password, "faceAndFp"-face+fingerprint, "faceAndPw"-face+password,
"faceAndCard"-face+card, "face"-face, "employeeNoAndPw"-employee No.+password,
"fpOrPw"-fingerprint or password, "employeeNoAndFp"-employee No.+fingerprint+password,
"faceAndFpAndCard"-face+fingerprint+card, "faceAndPwAndFp"-face+password+fingerprint,
"employeeNoAndFace"-employee No.+face, "faceOrfaceAndCard"-face or face+card,
"fpOrface"-fingerprint or face, "cardOrfaceOrPw"-card or face or password,
"cardOrFace"-card or face, "cardOrFaceOrFp"-card or face or fingerprint,
"cardOrFpOrPw"-card or fingerprint or password. The priority of the person authentication mode is higher than that of the card reader authentication mode*/
    "dynamicCode": "123456",
/*optional, string, dynamic permission code, this node is write-only*/
    "callNumbers": ["","",""],
/*optional, array of string, list of called numbers, the default rule is "X-X-X-X",
e.g., "1-1-1-401". This node is the extension of the node roomNumber. When
the number list is supported, you need to use this node to configure
parameters*/
    "floorNumbers": [1,2],
/*optional, array of int, floor No. list. This node is the extension of
floorNumber. When the number list is supported, you need to use this node to
configure parameters*/
    "numOfFace":0,
/*optional, int, number of linked face pictures. This node is read-only and if
it is not returned, it indicates that this function is not supported*/
    "numOfFP":0,
/*optional, int, number of linked fingerprints. This node is read-only and if
it is not returned, it indicates that this function is not supported*/
    "numOfCard":0,
/*optional, int, number of linked cards. This node is read-only and if it is
not returned, it indicates that this function is not supported*/
    "gender":"",
/*optional, string, gender of the person in the face picture: "male", "female",
"unknown"*/

```

```

    "PersonInfoExtends": [
        /*optional, array of object, extended fields for the additional person
        information. This node is used to configure the extended person information
        displayed on the device's UI. For MinMoe series facial recognition terminals,
        currently only one value node can be supported for displaying the employee No.
        and the node id is not supported*/
        {
            "id": 1,
            /*optional, int, extended ID of the additional person information, value range:
            [1,32]. It corresponds to the id in the message of the request URI /ISAPI/
            AccessControl/personInfoExtendName?format=json and is used to link the value of
            the node value and its name (the node name in the message of the request URI /
            ISAPI/AccessControl/personInfoExtendName?format=json). If the node id does not
            exist, the ID will start from 1 by default according to the array order*/
            "value": ""
        },
        /*optional, string, extended content of the additional person information*/
        ],
        "groupName": "test",
        /*optional, string. group name, range:[1,64]*/
        "age": 0,
        /*optional, integer, age, range:[0,120]*/
        "PatientInfos": {
            /*optional, object, patient infomation*/
            "deviceID": "test",
            /*optional, string, device number*/
            "admissionTime": "1970-01-01T00:00:00+08:00",
            /*optional, datetime, hospitalized date*/
            "chargeNurse": "test",
            /*optional, string, nurse in charge, range:[0,32]*/
            "chargeDoctor": "test",
            /*optional, string, doctor in charge, range:[0,32]*/
            "nursingLevel": "tertiary",
            /*optional, enumerate, nursing level*/
            "doctorsAdvice": "test",
            /*optional, string, advice from doctor, range:[0,128]*/
            "allergicHistory": "test"
        },
        /*optional, string, allergy, range:[0,128]*/
        },
        "TromboneRule": {
            /*optional, object, trombone rule*/
            "industryType": "builidings",
            /*optional, string, industry type*/
            "unitType": "indoor",
            /*optional, string, device type, indoor (idoor station), villa (villa outdoor
            station), confirm (double confirm), outdoor (outdoor station), fence (outer
            door station), doorbell (doorbell), manage (master station), acs (access
            control device), wardStation (ward extension), bedheadExtension (bedhead
            extension), bedsideExtension (bedside extension), terminal (terminal), netAudio
            (network audio), interactive (interactive terminal), amplifier (amplifier)*/
            "SIPVersion": "V10"
        },
        /*optional, string, private SIP version, range:[0,32]*/
        },
        "ESDType": "handAndFoot"
    ]
}

```

```
/*optional, enumerate, ESD detection type: handAndFoot (detect both hand and
foot), no (no detection), hand (detect hand), foot (detect foot)*/
    }]
}
}
```

### F.171 JSON\_UserInfoSearchCond

UserInfoSearchCond message in JSON format

```
{
  "UserInfoSearchCond": {
    "searchID": "",  

    /*required, string type, search ID, which is used to confirm the upper-level
    platform or system. If the platform or the system is the same one during two
    searching, the search history will be saved in the memory to speed up next
    searching*/
    "searchResultPosition": ,  

    /*required, integer32 type, the start position of the search result in the
    result list. When there are multiple records and you cannot get all search
    results at a time, you can search for the records after the specified position
    next time*/
    "maxResults": ,  

    /*required, integer32 type, maximum number of search results. If maxResults
    exceeds the range returned by the device capability, the device will return the
    maximum number of search results according to the device capability and will
    not return error message*/
    "EmployeeNoList": [{  

      /*optional, person ID list (if this node does not exist or is empty, it
      indicates searching for all person information)*/
      "employeeNo": ""  

      /*optional, string type, employee No. (person ID)*/
      },
      "fuzzySearch": "",  

      /*optional, string, key words for fuzzy search*/
      "userType": "normal",  

      /*optional, string, normal (normal user), visitor (visitor), blockList (person
      in blocklist), patient (patient), maintenance (maintenance people)*/
      "deviceIDList": [1, 2]
    /*optional, array, device ID list*/
    }
  }
}
```

### F.172 JSON\_UserRightHolidayGroupCfg

UserRightHolidayGroupCfg message in JSON format

```
{
  "UserRightHolidayGroupCfg": {
```

```

    "enable": ,
/*required, boolean, whether to enable: "true"-enable, "false"-disable*/
    "groupName": "",
/*required, string, holiday group name*/
    "holidayPlanNo": "",
/*required, string, holiday group schedule No.*/
    "operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal, "byOrg"-by
organization, "byTerminalOrg"-by terminal organization*/
    "terminalNoList": [ 1, 2, 3, 4 ],
/*optional, array, terminal ID list, this node is required when operation type
is "byTerminal" or "byTerminalOrg"*/
    "orgNoList": [ 1, 2, 3, 4 ]
/*optional, array, organization ID list, this node is required when operation
type is "byOrg" or "byTerminalOrg"*/
}
}

```

### F.173 JSON\_UserRightHolidayPlanCfg

JSON message about holiday schedule parameters of the access permission control

```

{
  "UserRightHolidayPlanCfg": {
    "enable": ,
/*required, boolean, whether to enable: "true"-enable, "false"-disable*/
    "beginDate": "",
/*start date of the holiday (device local time)*/
    "endDate": "",
/*end date of the holiday (device local time)*/
    "HolidayPlanCfg" : [{}],
/*holiday schedule parameters*/
    "id": ,
/*required, integer, time period No., which is between 1 and 8*/
    "enable": ,
/*required, boolean, whether to enable: "true"-enable, "false"-disable*/
    "TimeSegment": {
      "beginTime": "",
/*required, start time of the time period (device local time)*/
      "endTime": ""
/*required, end time of the time period (device local time)*/
    }
  ],
  "operateType": "byTerminal",
/*optional, string, operation type: "byTerminal"-by terminal, "byOrg"-by
organization, "byTerminalOrg"-by terminal organization*/
  "terminalNoList": [ 1, 2, 3, 4 ],
/*optional, array, terminal ID list, this node is required when operation type
is "byTerminal" or "byTerminalOrg"*/
  "orgNoList": [ 1, 2, 3, 4 ]
}

```

```
/*optional, array, organization ID list, this node is required when operation
type is "byOrg" or "byTerminalOrg"*/
    }
}
```

### F.174 JSON\_UserRightPlanTemplate

JSON message about schedule template configuration parameters of the access permission control

```
{
  "UserRightPlanTemplate": {
    "enable": ,
    /*required, boolean, whether to enable: "true"-enable, "false"-disable*/
    "templateName": "",
    /*required, string, template name*/
    "weekPlanNo": ,
    /*required, integer, week schedule No.*/
    "holidayGroupNo": "",
    /*required, string, holiday group No.*/
    "operateType": "byTerminal",
    /*optional, string, operation type: "byTerminal"-by terminal, "byOrg"-by
organization, "byTerminalOrg"-by terminal organization*/
    "terminalNoList": [ 1, 2, 3, 4 ],
    /*optional, array, terminal ID list, this node is required when operation type
is "byTerminal" or "byTerminalOrg"*/
    "orgNoList": [ 1, 2, 3, 4 ]
    /*optional, array, organization ID list, this node is required when operation
type is "byOrg" or "byTerminalOrg"*/
  }
}
```

### F.175 JSON\_UserRightWeekPlanCfg

JSON message about week schedule parameters of the access permission control

```
{
  "UserRightWeekPlanCfg": {
    "enable": ,
    /*required, boolean, whether to enable: "true"-enable, "false"-disable*/
    "WeekPlanCfg": [
      /*required, week schedule parameters*/
      "week": "",
      /*required, string, day of the week: "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday", "Sunday"*/
      "id": ,
      /*required, integer, time period No., which is between 1 and 8*/
      "enable": ,
      /*required, boolean, whether to enable: "true"-enable, "false"-disable*/
      "TimeSegment": {
        /*optional, array, time period ID list, this node is required when
operation type is "byTerminal" or "byTerminalOrg"*/
        "timePeriodList": [
          { "id": 1, "start": "00:00:00", "end": "01:00:00" },
          { "id": 2, "start": "01:00:00", "end": "02:00:00" },
          { "id": 3, "start": "02:00:00", "end": "03:00:00" },
          { "id": 4, "start": "03:00:00", "end": "04:00:00" },
          { "id": 5, "start": "04:00:00", "end": "05:00:00" },
          { "id": 6, "start": "05:00:00", "end": "06:00:00" },
          { "id": 7, "start": "06:00:00", "end": "07:00:00" },
          { "id": 8, "start": "07:00:00", "end": "08:00:00" }
        ]
      }
    ]
  }
}
```

```
        "beginTime": "",  
        /*required, start time of the time period (device local time)*/  
        "endTime": ""  
        /*required, end time of the time period (device local time)*/  
    }  
},  
    "operateType": "byTerminal",  
    /*optional, string, operation type: "byTerminal"-by terminal, "byOrg"-by  
organization, "byTerminalOrg"-by terminal organization*/  
    "terminalNoList": [ 1, 2, 3, 4 ],  
    /*optional, array, terminal ID list, this node is required when operation type  
is "byTerminal" or "byTerminalOrg"*/  
    "orgNoList": [ 1, 2, 3, 4 ]  
    /*optional, array, organization ID list, this node is required when operation  
type is "byOrg" or "byTerminalOrg"*/  
}  
}
```

### F.176 JSON\_Verification

JSON message about verification parameters of section password.

```
{  
    "Verification": {  
        "sectionNo": ,  
        /*requiried, integer, section No.*/  
        "passwordType": "",  
        /*optional, string, password types: "KeyA" (default), "KeyB"*/  
        "password": ""  
        /*optional, string, a hexadecimal key, which depends on the password type*/  
    }  
}
```

### F.177 XML\_AcsAbility

AcsAbility capability message in XML format

```
<AcsAbility version="2.0">  
    <channelControllerNo min="" max="" />  
    <!--required, lane controller range--&gt;<br/>    <doorNo min="" max="" />  
    <!--req , door No. rang or floor No. range--&gt;<br/>    <cardReaderNo min="" max="" />  
    <!--required, card reader No. range--&gt;<br/>    <maxCardNum></maxCardNum>  
    <!--required, supported card number--&gt;<br/>    <caseSensorNo min="" max="" />  
    <!--required, event trigger No.--&gt;<br/>    <gateOpenDirectionNum opt="1,2"/>
```

```

<!--required, the number of door opening directions (e.g., for the flap
barrier which has only one direction, the attribute "opt" should be set to 1;
for the swing barrier and the tripod turnstile which have two directions, the
attribute "opt" should be set to 2)-->
<DoorRelateCardReaderList>
    <!--optional, card reader No., which is linked with the door No. (it will
be returned only when the card reader has linked with card reader, otherwise it
will not be returned)-->
    <Action>
        <doorNo>1</doorNo>
        <cardReaderNo>1,2</cardReaderNo>
    </Action>
</DoorRelateCardReaderList>
<DoorStatusPlan>
    <!--required, door status schedule capability -->
    <WeekPlan>
        <!--required, weekly schedule capability -->
        <weekPlanNo min="" max="" />
        <!--required, weekly schedule No. range -->
        <maxDaySegment>8</maxDaySegment>
        <!--required, supported daily time segment number -->
        <status opt="invalid,sleep,alwaysopen,alwaysclose,normal"/>
        <!--required, status value range -->
        <verifyType
opt="invalid,sleep,swipecard,swipecardandpassword,swipecardorpasswd,fingerPrint,
fingerPrintAndPasswd,fingerPrintorCard,fingerPrintAndCard,fingerPrintAndCardAndP
asswd,faceOrFpOrCardOrPw,faceAndFingerPrint,faceAndPassword,faceAndCard,face,emp
loyeeNoAndPassword,fingerPrintOrPassword,employeeNoAndFp,employeeNoAndFpAndPw,fa
ceAndFpAndCard,faceAndPwAndFp,employeeNoAndface,cardOrFace,cardOrFaceOrFp,cardOr
FpOrPw"/>
        <!--required, authentication method range -->
        <TimeAccuracy>
            <!--required, time accuracy -->
            <hour>enable</hour>
            <minute>enable</minute>
            <second>enable</second>
        </TimeAccuracy>
    </WeekPlan>
    <HolidayPlan>
        <!--required, holiday schedule -->
        <holidayPlanNo min="" max="" />
        <!--required, holiday schedule No. range -->
        <maxDaySegment>8</maxDaySegment>
        <!--required, supported daily time segment number -->
        <TimeAccuracy>
            <!--required, time accuracy-->
            <hour>enable</hour>
            <minute>enable</minute>
            <second>enable</second>
        </TimeAccuracy>
    </HolidayPlan>
    <HolidayGroup>

```

```

<!--required, holiday group capability-->
<holidayGroupNo min="" max="" />
<!--required, holiday group No. range -->
<holidayGroupName min="" max="" />
<!--required, holiday group name length -->
<maxHolidayPlanNum></maxHolidayPlanNum>
<!--required, max. holiday schedule number for the holiday group -->
</HolidayGroup>
<PlanTemplate>
    <!--required, schedule template capability -->
    <templateNo min="" max="" />
        <!--optional, range of schedule template No.-->
    <templateName min="" max="" />
    <!--required, schedule template name length -->
    <maxHolidayGroupNum></maxHolidayGroupNum>
    <!--required, max. holiday group number for the schedule template -->
</PlanTemplate>
<supportLocalController>enable</supportLocalController>
<!--required, support distributed access controller-->
</DoorStatusPlan>
<CardReaderVerifyTypePlan>
    <!--required, card reader authentication schedule capability -->
    <WeekPlan>
        <!--required, weekly schedule capability -->
        <weekPlanNo min="" max="" />
        <!--required, weekly schedule No. range -->
        <maxDaySegment>8</maxDaySegment>
        <!--required, supported daily time segment number -->
        <status opt="invalid,sleep,alwaysopen,alwaysclose,normal" />
        <!--required, status value range -->
        <verifyType
opt="invalid,sleep,swipecard,swipecardandpassword,swipecardorpasswd,fingerPrint,
fingerPrintAndPasswd,fingerPrintOrCard,fingerPrintAndCard,fingerPrintAndCardAndP
asswd,fingerPrintorCard,fingerPrintAndCard,fingerPrintAndCardAndPasswd,faceOrFpO
rCardOrPw,faceAndFingerPrint,
faceAndPassword,faceAndCard,face,employeeNoAndPassword,fingerPrintOrPassword,emp
loyeeNoAndFp,employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndf
ace,employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndface,faceO
rFaceAndCard,fingerPrintOrFace,swipecardOrFaceOrPw,cardOrFace,cardOrFaceOrFp,car
dOrFpOrPw"/>
        <!--required, verification mode range: invalid, sleep, card, card and
password, card or password, fingerprint, fingerprint and password, fingerprint
or card, fingerprint and card, fingerprint and card and password (no order),
face or fingerprint or card or password, face and fingerprint, face and
password, face and card, face, employee No. and password, fingerprint or
password, employee No. and fingerprint, employee No. and fingerprint and
password, face and fingerprint and card, face and password and fingerprint,
employee No. and face, employee No. and fingerprint and password, face and
fingerprint and card, face and password and fingerprint, employee No. and face,
face or face and card, fingerprint or face, card or face or password, card or
face, card or face or fingerprint-->
        <purePwdVerifyEnable><!--optional, boolean, whether the device supports

```

```
opening the door only by password: true-yes, this node is not returned-no--></purePwdVerifyEnable>
    <!--For opening the door only by password: 1. The password in "XXX or password" in the authentication mode refers to the person's password (the value of the node password in JSON_UserInfo); 2. The device will not check the duplication of the password, and the upper platform should ensure that the password is unique; 3. The password cannot be added, deleted, edited, or searched for on the device locally-->
    <TimeAccuracy>
        <!--required, time accuracy -->
        <hour>enable</hour>
        <minute>enable</minute>
        <second>enable</second>
    </TimeAccuracy>
</WeekPlan>
<HolidayPlan>
    <!--required, holiday schedule -->
    <holidayPlanNo min="" max="" />
    <!--required, holiday schedule No. range -->
    <maxDaySegment>8</maxDaySegment>
    <!--required, supported daily time segment number -->
    <TimeAccuracy>
        <!--required, time accuracy -->
        <hour>enable</hour>
        <minute>enable</minute>
        <second>enable</second>
    </TimeAccuracy>
</HolidayPlan>
<HolidayGroup>
    <!--required, holiday group capability -->
    <holidayGroupNo min="" max="" />
    <!--required, holiday group No. range -->
    <holidayGroupName min="" max="" />
    <!--required, holiday group name length -->
    <maxHolidayPlanNum></maxHolidayPlanNum>
    <!--required, max. holiday schedule number for holiday group -->
</HolidayGroup>
<PlanTemplate>
    <!--required, schedule template capability -->
    <templateNo min="" max="" />
    <!--optional, range of schedule template No.-->
    <templateName min="" max="" />
    <!--required, schedule template name length -->
    <maxHolidayGroupNum></maxHolidayGroupNum>
    <!--required, max. holiday group number for schedule template -->
</PlanTemplate>
<supportLocalController>enable</supportLocalController>
    <!--required, support distributed access controller-->
</CardReaderVerifyTypePlan>
<CardRightPlan>
    <!--required, card permission schedule capability -->
    <WeekPlan>
```

```

<!--required, weekly schedule capability -->
<weekPlanNo min="" max="" />
<!--required, weekly schedule No. range -->
<maxDaySegment>8</maxDaySegment>
<!--required, supported daily time segment number -->
<status opt="invalid,sleep,alwaysopen,alwaysclose,normal" />
<!--required, status value range -->
<verifyType opt="invalid,sleep,swipecard,swipecardandpassword" />
<!--required, authentication method range -->
<TimeAccuracy>
    <!--required, time accuracy -->
    <hour>enable</hour>
    <minute>enable</minute>
    <second>enable</second>
</TimeAccuracy>
</WeekPlan>
<HolidayPlan><!--required, holiday schedule -->
    <holidayPlanNo min="" max="" /><!--required, holiday schedule No. range -->
    >
        <maxDaySegment>8</maxDaySegment>
        <!--required, supported daily time segment number -->
        <TimeAccuracy>
            <!--required, time accuracy -->
            <hour>enable</hour>
            <minute>enable</minute>
            <second>enable</second>
        </TimeAccuracy>
    </HolidayPlan>
    <HolidayGroup>
        <!--required, holiday group capability-->
        <holidayGroupNo min="" max="" />
        <!--required, holiday group No. range -->
        <holidayGroupName min="" max="" />
        <!--required, holiday group name length -->
        <maxHolidayPlanNum></maxHolidayPlanNum>
        <!--required, max. holiday schedule number for holiday group -->
    </HolidayGroup>
    <PlanTemplate>
        <!--required, schedule template capability -->
        <templateNo min="" max="" />
            <!--optional, range of schedule template No.-->
        <templateName min="" max="" />
        <!--required, schedule template name length -->
        <maxHolidayGroupNum></maxHolidayGroupNum>
        <!--required, max. holiday group number for schedule template -->
    </PlanTemplate>
    <supportLocalController>enable</supportLocalController>
        <!--required, support distributed access controller-->
</CardRightPlan>
<Door>
    <!--required, door parameters capaility -->
    <doorName min="" max="" />

```

```

<!--required, door name length -->
<magneticMode opt="alwaysclose,alwaysopen"/>
<!--required, door magnetic type -->
<openButtonMode opt="alwaysclose,alwaysopen"/>
<!--required, exit button type-->
<openDuration min="" max="" />
<!--required, door opening duration range, unit: second -->
<disabledOpenDuration min="" max="" />
<!--required, disabled card opening door duration range, unit: second)-->
<magneticAlarmTimeout min="" max="" />
<!--required, magnetic detection overtime alarm time, unit: second, 0
indicates not to alarm. -->
<doorLock>enable</doorLock>
<!--required, whether support locking door when door closed. -->
<leaderCard>enable</leaderCard>
<!--required, whether to enable first card opening door -->
<stressPassword min="" max="" />
<!--required, duress password length -->
<superPassword min="" max="" />
<!--required, super password length -->
<unlockPassword min="" max="" />
<!--optional, unlocking password length -->
<leaderCardMode opt="close,alwaysopen,authorized"/>
<!--required, first card mode-->
<useLocalController>enable</useLocalController>
<!--required, whether the door is connected to distributed access
controller-->
<localControllerID min="" max="" />
<!--required, distributed access controller No.-->
<localControllerDoorNumber min="" max="" />
<!--required, distributed access controller door No.-->
<localControllerStatus opt="offline,netOnline,authorized"/>
<!--required, distributed access controller online status-->
<lockInputCheck>enable</lockInputCheck>
<!--required, whether to enable door lock input check (1 byte, 0- disable,
1- enable, default to disable)-->
<lockInputType opt="NormallyClose,NormallyOpen"/>
<!--required, door lock input type (1 byte, 0- normally closed, 1- normally
open, default to normally closed)-->
<doorTerminalMode opt="PreventCutShort,Normal"/>
<!--required, door related terminal operating mode (1 byte, 0- anti-cut &
short-circuit, 1- normal, default to anti-cut & short-circuit)-->
<openButton>enable</openButton>
<!--required, whether to enable door button (1 byte, 0- yes, 1- no, default
to yes)-->
</Door>
<DoorStatusPlan>
    <!--required, door status schedule parameters -->
    <enable>true</enable>
</DoorStatusPlan>
<Group>
    <!--required, group parameters capability -->

```

```

<ValidCfg>
    <!--required, validate capability -->
    <TimeAccuracy>
        <!--required, time accuracy -->
        <year>enable</year>
        <month>enable</month>
        <day>enable</day>
        <hour>enable</hour>
        <minute>enable</minute>
        <second>enable</second>
    </TimeAccuracy>
    <timeType opt="local,UTC"/>
        <!--optional, time type: "local"-device local time (default), "UTC"-UTC
time>
    </ValidCfg>
    <groupName min="" max="" />
        <!--required, group name length -->
        <groupNo min="" max="" /><!--required, group No. range. If this node cannot
be parsed or is not returned, it will be set to the default value-->
</Group>
<MultiCard>
    <!--required, multi-card capability -->
    <swipeIntervalTimeout min="" max="" />
        <!--required, multi-card swiping interval overtime, unit: second -->
    <maxMultiCardGroupNum, min="1", max="20"></maxMultiCardGroupNum>
        <!--required, max. multi-card group number >
    <maxGroupCombinationNum></maxGroupCombinationNum>
        <!--required, max. group number for a multi-card group -->
    <remoteOpenDoor>enable</remoteOpenDoor>
        <!--required, supports remote door opening authentication method -->
    <offlineVerifyMode>enable</offlineVerifyMode>
        <!--required, supported offline control panel authentication mode (super
password replaces remote door opening control) -->
</MultiCard>
<Card>
    <!--required, card parameters capability -->
    <cardNo min="" max="" />
        <!--required, card No. length -->
    <modifyParamType
opt="cardvalid,validtime,cardtype,doorright,leadercard,swipenum,group,password,
rightplan, swipednum, employeno, name, departmentNo, schedulePlanNo,
schedulePlanType,roomNo,simNo,floorNo,userType"/>
        <!--required,edit separately --> opt="cardvalid- card valid or not,
validtime- expiry date, cardtype- card type, doorright- door permission,
leadercard- first card, swipenum- max. card swiping times, group- group,
password- card password,,rightplan- card permission schedule,
swipednum- card swiped times, employeno- employee No., name-Name,
departmentNo-Apartment No., schedulePlanNo-Schedule No., schedulePlanType-
Schedule Type-->
    <cardValid>enable</cardValid>
    <timeRangeBegin>
        <!--optional, start time that can be configured by beginTime and endTime.

```

```

If this node is not returned by the capability, the start time that can be
configured is 1970-01-01T00:00:00 by default-->
    </timeRangeBegin>
    <timeRangeEnd>
        <!--optional, end time that can be configured by beginTime and endTime.
If this node is not returned by the capability, the end time that can be
configured is 2037-12-31T23:59:59 by default-->
    </timeRangeEnd>
    <cardValidUnit opt="day,hour,minute,second">
        <!--required, accuracy of card expiry date (if device supports correcting
to minute, opt="minute"), if this node is not returned, the default accuracy is
day (opt="day")>
    </cardValidUnit>
    <!--required, whether the card is valid-->
    <!--required, card type-->
    <cardType opt="normalcard,disabledcard,blacklistcard,nightwatchcard,
stresscard,supercard,guestcard,mastercard,staffcard,normalopencard,cleancard,sta
ndbycard, unlockcard"/>
    <doorRight>enable</doorRight>
    <!--required, door permission-->
    <leaderCard>enable</leaderCard>
    <!--required, whether to enable the first card? -->
    <swipeNum min="" max="" />
    <!--required, max. card swiping number, o indicates no limit-->
    <maxBelongGroup></maxBelongGroup>
    <!--required, max. group number belonged to -->
    <cardPassword min="" max="" />
    <!--required, card password-->
    <doorRightPlanNum></doorRightPlanNum>
    <!--required, max. schedule template number for a single door -->
    <swipeTime>enabled</swipeTime>
    <!--required, swiping times -->
    <onlyPasswdOpen opt="true,false"/>
    <!--optional, whether to support password opening door, invalid currently --
>
    <roomNumber min="" max="" />
    <!--optional, Room No.-->
    <floorNumber min="" max="" />
    <!--optional, Floor number-->
    <employeeNo min="" max="" />
    <!--optional, employee No.-->
    <name min="" max="" /></name>
    <!--required, name (if device returns this node, you can get and set the
linked user name of the card by calling card parameter API directly, so there
is no need to API NET_DVR_SET_CARD_USERINFO_CFG and
NET_DVR_GET_CARD_USERINFO_CFG)-->
    <departmentNo min="" max="" />
    <!--optional, department No.-->
    <schedulePlanNo min="" max="" />
    <!--optional, shift schedule-->
    <schedulePlanType opt="personal,department"/>
    <!--optional, shift schedule type-->

```

```

<lockID min="" max="" />
<!--required, lock ID-->
<roomCode min="" max="" />
<!--required, room code-->
<cardRight
opt="lowPowerAlarm,openDoorSound,customCardLimit,normalOpen,openLockedDoor,keepW
atch"/>
    <!--required, card permission-->
    <supportLocalController>enable</supportLocalController>
    <!--required, support distributed access controller-->
    <roomNumber min="" max="" /></roomNumber><!--required, room No.>
    <floorNumber min="" max="" /></floorNumber><!--required, floor No.>
    <SIMNum min="" max="" /></SIMNum><!--required, mobile phone number>
    <isSupportCardModify>true</isSupportCardModify>
        <!--required, support downloading when card parameters changed (for video
intercom device only, by default, this function is supported by all access
control devices)>
</Card>
<AntiSneak>
    <!--required, anti-passback capability-->
    <startCardReaderNo>enable</startCardReaderNo>
    <!--required, anti-passback card reader No. configuration -->
    <maxSneakPath></maxSneakPath>
    <!--required, max. anti-passback follow-up card reader number-->
</AntiSneak>
<MultiDoorInterlock>
    <!--required, multi-door interlocking parameters -->
    <maxMultiDoorInterlockGroup></maxMultiDoorInterlockGroup>
    <!--required, max. multi-door interlocking group number -->
    <maxInterlockDoorNum></maxInterlockDoorNum>
    <!--required, max. interlocked door number for one multi-door interlocking
group -->
</MultiDoorInterlock>
<AcsWorkStatus>
    <!--required, access controller working status parameters -->
    <doorLogicalStatus>enable</doorLogicalStatus>
    <!--required, door logic status -->
    <doorStatus opt="alwaysopen,alwaysclose,normal"/>
    <!--required, door status parameters -->
    <magneticStatus>enable</magneticStatus>
    <!--required, door magnetic status parameters -->
    <relayStatus>enable</relayStatus>
    <!--required, relay status-->
    <caseSensorStatus>enable</caseSensorStatus>
    <!--required, case trigger status-->
    <BatteryVoltage>enable</BatteryVoltage>
    <!--required, battery voltage value -->
    <BatteryLowVoltage>enable</BatteryLowVoltage>
    <!--required, battery low voltage detection -->
    <PowerSupplyStatus>enable</PowerSupplyStatus>
    <!--required, device power supply status-->
    <multiDoorInterlockStatus>enable</multiDoorInterlockStatus>

```

```

<!--required, multi-door interlocking status parameters-->
<antiSneakStatus>enable</antiSneakStatus>
<!--required, anti-passback status parameters-->
<hostAntiDismantleStatus>enable</hostAntiDismantleStatus>
<!--required, control ler tamper ?proof status-->
<indicatorLightStatus>enable</indicatorLightStatus>
<!--required, Supports indicator status-->
<cardReaderOnlineStatus>enable</cardReaderOnlineStatus>
<!--required, card reader connection status -->
<cardReaderAntiDismantleStatus>enable</cardReaderAntiDismantleStatus>
<!--required, card reader tamper-proof status -->
    <cardReaderVerifyMode opt="invalid,sleep,swipecard,swipecardandpassword,
swipecardorpasswd, fingerPrint,fingerPrintAndPasswd,fingerPrintor
Card,fingerPrintAndCard,fingerPrintAndCardAndPasswd,faceOrFpOrCardOrPw,
faceAndFingerPrint,faceAndPassword,faceAndCard,face,employeeNoAndPassword,finger
PrintOrPassword,employeeNoAndFp,
employeeNoAndFpAndPw,faceAndFpAndCard,faceAndPwAndFp,employeeNoAndface,faceOrfac
eAndCard,fingerPrintOrFace,swipecardOrFaceOrPw,"/>
        <!--required, supported card reader authentication modes: 0-invalid, 1-
card, 2-card+password, 3-card, 4-card/password, 5-fingerprint, 6-fingerprint
+password, 7-fingerprint/card, 8-fingerprint_card, 9-fingerprint_card+password,
10-face/fingerprint/card/password, 11-face+fingerprint, 12-face+password, 13-
face+card, 14-face, 15-employee No.+password, 16-fingerprint/password, 17-
employee No.+fingerprint, 18-employee No.+fingerprint+password, 19-face
+fingerprint+card, 20-face+password+fingerprint, 21-employee No.+face, 22-face/
face+card, 23-fingerprint/face, 24-card/face/password-->
            <setupAlarmStatus>enable</setupAlarmStatus>
            <!--required, zone arming status -->
            <alarmInStatus>enable</alarmInStatus>
            <!--required, alarm input status -->
            <alarmOutStatus>enable</alarmOutStatus>
            <!--required, alarm output status -->
            <cardNum>enable</cardNum>
            <!--required, added card number -->
<fireAlarmStatus opt="normal,shortCircuit,break"/>
    <!--required, support fire alarm status-->
    <supportLocalController>enable</supportLocalController>
    <!--required, support distributed access controller-->
    <batteryChargeStatus opt="InCharge,NotCharge"/>
    <!--required, battery status: InCharge-Charging, NotCharge-Uncharged>
    <masterChannelControllerStatus>enable</masterChannelControllerStatus>
    <!--required, supports online status of main lane controller-->
    <slaveChannelControllerStatus>enable</slaveChannelControllerStatus>
    <!--required, supports online status of sub-lane controller-->
    <antiSneakServerStatus opt="disable,normal,disconnect"/>
        <!--optional, anti-passing back server status: "disable"-disabled,
"normal"-normal, "disconnect"-disconnected-->
        <whiteFaceNum>enable</whiteFaceNum>
        <!--required, supports the parameters of face picture quantity in
allowlist-->
        <blackFaceNum>enable</blackFaceNum>
        <!--required, supports the parameters of face picture quantity in
allowlist-->

```

```

blocklist-->
  </AcsWorkStatus>
  <CaseSensor>
    <!--required, event trigger parameters capability -->
    <triggerHostBuzzer>enable</triggerHostBuzzer>
    <!--required, trigger controller buzzer -->
    <triggerCardReaderBuzzer>enable</triggerCardReaderBuzzer>
    <!--required, trigger card reader buzzer -->
    <triggerAlarmOut>enable</triggerAlarmOut>
    <!--required, trigger alarm output -->
  <triggerDoorOpen>enable</triggerDoorOpen>
    <!--required, support triggered open door by ID-->
    <triggerAlarmOutClose>enable</triggerAlarmOutClose>
    <!--required, support disable triggered alarm input-->
    <triggerAlarmInSetup>enable</triggerAlarmInSetup>
    <!--required, support triggered arming region arming-->
    <triggerAlarmInClose>enable</triggerAlarmInClose>
    <!--required, support triggered arming region disarming-->
  </CaseSensor>
  <CardReaderCfg>
    <!--required, reader parameters capability-->
    <!--required, supported reader type-->
    <cardReaderType opt="DS-K110XM/MK/C/CK, DS-K192AM/AMP, DS-K192BM/BMP, DS-K182AM/AMP, DS-K182BM/BMP, DS-K182AMF/ACF,
      Wiegand or RS485 offline, DS-K1101M/MK, DS-K1101C/CK, DS-K1102M/MK/M-A, DS-K1102C/CK, DS-K1103M/MK,
      DS-K1103C/CK, DS-K1104M/MK, DS-K1104C/CK, DS-K1102S/SK/S-A, DS-K1102G/GK, DS-K1100S-B, DS-K1102EM/EMK,
      DS-K1102E/EK, DS-K1200EF, DS-K1200MF, DS-K1200CF, DS-K1300EF, DS-K1300MF, DS-K1300CF, DS-K1105E,
      DS-K1105M, DS-K1105C, DS-K182AMF, DS-K196AMF, DS-K194AMP, DS-K1T200EF/EF-C/MF-MF-C/CF/CF-C,
      DS-K1T300EF/EF-C/MF-MF-C/CF/CF-C"/>
    <okLedPolarity op="cathode,anode"/>
    <!--required,OK LED polarity-->
    <errorLedPolarity op="cathode,anode"/>
    <!--required,ERROR LED polarity-->
    <buzzerLedPolarity op="cathode,anode"/>
    <!--required, buzzer polarity -->
    <swipeInterval min="" max="" />
    <!--required, time interval of duplicated authentication, unit: second -->
    <pressTimeout min="" max="" />
    <!--required, key pressing overtime, unit: second -->
    <enableFailAlarm>enable</enableFailAlarm>
    <!--required, whether to enable authentication failure over times alarm configuration-->
    <maxReadCardFailNum min="" max="" />
    <!--required, max. times of authentication failure -->
    <enableTamperCheck>enable</enableTamperCheck>
    <!--optional, whether to support anti-tamper check-->
    <offlineCheckTime min="" max="" />
    <!--optional, offline check time, unit:s-->

```

```
<fingerPrintCheckLevel  
opt="1/10,1/100,1/1000,1/10000,1/100000,1/1000000,1/10000000,1/100000000,3/100,3  
/1000,  
  
3/10000,3/100000,3/1000000,3/10000000,3/100000000,Automatic Normal,Automatic  
Secure,Automatic More Secure"/>  
    <!--optional, fingerprint recognition level-->  
    <useLocalController>enable</useLocalController>  
    <!--required, whether door is connected to distributed access controller-->  
    <localControllerID min="" max="" />  
    <!--optional, distributed access controller No.-->  
    <localControllerReaderID min="" max="" />  
    <!--optional, ID of distributed access controller card reader-->  
    <cardReaderChannel opt="Wiegand/Offline,RS485A,RS485B"/>  
    <!--opt card reader communication channel No.-->  
    <fingerPrintImageQuality min="1" max="8"/>  
    <!--optional,fingerprint picture quality-->  
    <fingerPrintContrastTimeOut min="0" max="20"/>  
    <!--optional,fingerprint comparison overtime, 0 - infinite, that is 0xff-->  
    <fingerPrintRecognizeInterval min="0" max="10"/>  
    <!--optional,time interval of fingerprint continuous recognition, 0- no  
delay, that is 0xff-->  
        <fingerPrintMatchFastMode min="0" max="5"/>  
        <!--optional,fingerprint fast matching mode, 0- auto, that is 0xff-->  
        <fingerPrintModuleSensitive min="1" max="8"/>  
        <!--optional,fingerprint module sensitivity-->  
        <fingerPrintModuleLightCondition opt="outdoor,indoor"/>  
        <!--optional,light condition of fingerprint module-->  
        <faceMatchThresholdN min="0" max="100"/>  
        <!--optional,face 1:N matching threshold-->  
        <faceQuality min="0" max="100"/>  
        <!--optional,face picture quality-->  
        <faceRecognizeTimeOut min="0" max="20"/>  
        <!--optional,face recognition overtime, 0 - infinite, that is 0xff-->  
        <faceRecognizeInterval min="0" max="10"/>  
        <!--optional,time interval of face continuous recognition, 0- no delay,  
that is 0xff-->  
            <cardReaderFunction opt="fingerPrint,face,fingerVein"/>  
            <!--optional,card reader types-->  
            <cardReaderDescription min="1" max="32"/>  
            <!--optional,card reader description-->  
            <faceImageSensitometry1 min="0" max="65535"/>  
            <!--optional, face picture exposure-->  
            <livingBodyDetect opt="disable,enable"/>  
            <!--optional, face detection-->  
            <faceMatchThreshold1 min="0" max="100"/>  
            <!--optional,Face 1:1 matching threshold-->  
            <buzzerTime min="0" max="5999"/>  
            <!--optional, buzzing time-->  
            <faceMatch1SecurityLevel opt="normal, more secure, extremely secure"/>  
            <!--optional, face picture 1:1 security level: 1-normal, 2-high, 3-higher-->  
>
```

```

<faceMatchNSecurityLevel opt="0,1,2"/>
  <!--optional, face picture 1:N security level: 1-normal, 2-high, 3-higher-->
>
<envirMode opt="normal, more secure, extremely secure"/>
  <!--optional, face recognition environment mode: 0-invalid, 1-indoor, 2-other -->
<liveDetLevelSet opt="0,1,2,3"/>
  <!--optional, set live face detection threshold level: 0-invalid, 1-low, 2-medium, 3-high-->
<liveDetAntiAttackCntLimit min="0"max="255"/>
  <!--optional, max. live face detection failed attempts-->
<enableLiveDetAntiAttack opt="0,1,2"/>
  <!--optional, whether enable locking face: 0-invalid, 1-disabled, 2-enabled-->
<fingerPrintCapacity min="" max="" />
  <!--ro, optional, xs:integer, fingerprint capacity-->
<fingerPrintNum min="" max="" />
  <!--ro, optional, xs:integer, the number of existed fingerprints-->
<enableFingerPrintNum opt="true"/>
  <!--ro, optional, xs:boolean, enable fingerprint capacity or not (when it is "true", fingerPrintCapacity and fingerPrintNum are valid)-->
<envirMode opt="0,1,2"></envirMode>
  <!--optional, environment mode of face recognition, 0-invalid, 1-indoor, 2-other-->
<liveDetLevelSet opt="0,1,2,3"></liveDetLevelSet>
  <!--optional, set live face detection security level, 0-invalid, 1-normal, 2-high, 3-higher-->
<liveDetAntiAttackCntLimit min="0"max="255">/liveDetAntiAttackCntLimit>
  <!--optional, maximum failed attempts-->
<enableLiveDetAntiAttack opt="0,1,2">
  <!--optional, enable locking face, 0-invalid, 1-disable, 2-enable-->
</enableLiveDetAntiAttack>
<faceContrastMotionDetLevel opt="low,middle,high"/><!--optional, motion detection level during face picture comparison: low, middle, high-->
<dayFaceMatchThresholdN min="0" max="100"/><!--optional, 1:N face picture comparison threshold in day-->
<nightFaceMatchThresholdN min="0" max="100"/><!--optional, 1:N face picture comparison threshold at night-->
<faceRecognizeEnable opt="true,false,multi"/><!-optional, whether to enable facial recognition: "true"-yes (one face), "false"-no, "multi"-yes (multiple faces)-->
<supportDelFPByID opt="true"/>
  <!--ro, optional, xs:boolean, whether the fingerprint module supports deleting fingerprint by fingerprint ID: "true"-yes, "false"-no-->
<defaultVerifyMode
opt="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27"/>
  <!--ro, optional, card reader authentication mode (factory default): 1-sleeping mode, 2-card swiping + password, 3-card swiping, 4-card swiping or password, 5-fingerprint, 6-fingerprint + password, 7-fingerprint or card swiping, 8-fingerprint + card swiping, 9-fingerprint + card swiping + password, 10-face or fingerprint or card swiping or password, 11-face + fingerprint, 12-face + password, 13-face + card swiping, 14-face, 15-employee ID + password, 16-

```

```

fingerprint or password, 17-employee ID + fingerprint, 18-employee ID +
fingerprint + password, 19-face + fingerprint + card swiping, 20-face +
password + fingerprint, 21-employee ID + face, 22-face or face + card swiping,
23-fingerprint or face, 24-card swiping or face or password, 25-card or face,
26-card or face or fingerprint, 27-card or fingerprint or password-->
    <fingerPrintCapacity min="" max="" /><!--ro, optional, xs: integer,
fingerprint capacity, this node is valid only when enableFingerPrintNum is
"true"-->
    <fingerPrintNum min="" max="" /><!--ro, optional, xs: integer, number of
existing fingerprints, this node is valid only when enableFingerPrintNum is
"true"-->
    <enableFingerPrintNum opt="true" /><!--ro, optional, xs: boolean, whether to
enable fingerprint capacity-->
    <blackFaceMatchThreshold min="0" max="100" /><!--optional, face picture
comparison threshold in blocklist-->
</CardReaderCfg>
<AcsUpgrade><!--required, upgrade capability of access control device-->
    <hostUpgrade>
        <!--required, whether to support upgrading main module-->
    </hostUpgrade>
    <cardReaderUpgrade>
        <!--required, whether to support upgrading card reader-->
    </cardReaderUpgrade>
    <localControllerUpgrade>
        <!--required, whether to support upgrading distributed access controller-->
    </localControllerUpgrade>
    <channelControllerUpgrade>
        <!--required, whether to support upgrading lane access controller-->
    </channelControllerUpgrade>
    <extensionModuleUpgrade>
        <!--required, whether to support upgrading extension module-->
    </extensionModuleUpgrade>
    <smartLockUpgrade>
        <!--required, whether to support upgrading smart lock-->
    </smartLockUpgrade>
    <cardReaderFPAlgorithmUpgrade>
        <!--required, whether to support upgrading fingerprint algorithm program
of fingerprint module-->
    </cardReaderFPAlgorithmUpgrade>
    <outdoorModules>
        <!--optional, whether to support upgrading the modules of door station,
if not support, this node will not be returned-->
    </outdoorModules>
    <modules opt="keybord,display,button,card,signal" />
        <!--opt, supported module type, "keybord"-keypad module, "display"-display
module,"button"-nametag module. "card"-card reader, "signal"-indicator module,
if not support, this node will not be returned-->
    </AcsUpgrade>
    <clearAcsParam
opt="doorstatusweekplan,cardreaderverifyweek,cardrightweekplan,doorstatusholiday
plan,cardreaderverifyholidayplan,cardrightholidayplan,doorstatusholidayplan,door

```

```

statusholidaygroup, cardreaderverifyholidaygroup, cardrightplantemplate, doorstatus
plantemolate, cardreaderverifyplantemplate, card, group, antisneak, eventandCardLinka
ge, cardPasswdOpendoor, personStatistics, blackListPicture, IDBlackList"/>
    <!--required, supported parameters clearing option -->
<ACSClearParam>
    <!--required, extend clear access control host parameter node-->
    <localControllerID min="" max="" />
    <!--optional, distributed access controller No.-->
</ACSClearParam>
<MultiHostAntiSneak>
    <!--required, over-controllers anti-passback -->
    <startAntiSnealHost opt="true,false"/>
    <!--optional, whether to enable anti-passback controller -->
    <antiSnealHostNum min="" max="" />
    <!--required, controller number for anti-passback controller group -->
    <ReadersCfg>
        <!--required, over-controllers anti-passback card reader parameters -->
        <maxRouteGroupNum></maxRouteGroupNum>
        <!--required, max. path number -->
        <oneRouteReadersNum min="" max="" />
        <!--required, follow-up card reader number for each path -->
    </ReadersCfg>
</MultiHostAntiSneak>

<AcsHostCfg>
    <!--optional, access control settings capability -->
    <enableRS485Backup opt="true,false"/>
    <!--required, whether to support downstream RS485 communication backup -->
    <showCapPic opt="true,false"/>
    <!--optional,whether to support displaying captured picture on LCD
screen-->
    <showCardNo opt="true,false"/>
    <!--optional,whether to support displaying card No. on LCD screen-->
    <showUserInfo opt="true,false"/>
    <!--optional,whether to support displaying user information on LCD
screen-->
    <overlayUserInfo opt="true,false"/>
    <!--optional,Whether to overlay user information on the captured picture-->
>
    <voicePrompt opt="true,false"/>
    <!--optional,Whether to support sound prompt-->
    <uploadCapPic opt="true,false"/>
    <!--optional, Whether to support uploading picture after capturing-->
    <saveCapPic opt="true,false"/>
    <!--optional,Whether to support saving captured picture-->
    <inputCardNo opt="true,false"/>
    <!--optional, whether supports inputting card No. by button-->
    <wifiDetect opt="true,false"/>
    <!--optional, whether supports enabling Wi-Fi probe-->
    <enable3G4G opt="true,false"/>
    <!--optional, enable 3G/4G-->
    <protocol opt="Private,OSDP"/>

```

```

<!--optional, card reader communication protocol type: "Private"-private
protocol (default), "OSDP"-OSDP protocol-->
</AcsHostCfg>

<EventLinkage>
    <!--required, event card linkage-->
    <maxEventNum></maxEventNum>
    <!--required, max. event linkage number supported by the device -->
    <supportMode opt="Event,CardNo,MAC,EmployeeNo"/>
    <!--required, supported linkage method, "Event"-event linkage, "CardNo"-
Card No. linkage, "MAC"-MAC address linkage, "EmployeeNo"-Employee No. (person
ID)-->
    <isSupportRecordVideo opt="true,false"/>
    <!--required, whether supports recording linkage-->
    <supportLocalController>enable</supportLocalController>
    <!--required, support distributed access controller-->
    <isSupportAlarmOutClose opt="true,false"/>
    <!--required, whether supports disabling linked alarm output-->
    <isSupportAlarmInSetup opt="true,false"/>
    <!--required, whether supports arming linked zone-->
    <isSupportAlarmInClose opt="true,false"/>
    <!--required, whether supports disarming linked zone-->
    <isSupportMainDevStopBuzzer opt="true,false"/>
    <!--required, whether supports stopping buzzing by access controller-->
    <isSupportReaderStopBuzzer opt="true,false"/>
    <!--required, whether supports stopping buzzing by linked card reader-->
    <audioDisplayMode opt="Close,SinglePlay,CyclePlay"/>
    <!--required, linked audio prompt mode: "Close"-disable, "SinglePlay"-play
once, "CyclePlay"-loop playing-->
    <audioDisplayID min="1" max="32"/>
    <!--required, linked audio prompt mode: "Close"-disable, "SinglePlay"-play
once, "CyclePlay"-loop playing-->
    <isNotSupportOpenDoor>
        <!--optional, whether the opening door linkage is not supported-->
    </isNotSupportOpenDoor>
    <isNotSupportCloseDoor>
        <!--optional, whether the closing door linkage is not supported-->
    </isNotSupportCloseDoor>
    <isNotSupportNormalOpen>
        <!--optional, whether the remaining door open is not supported-->
    </isNotSupportNormalOpen>
    <isNotSupportNormalClose>
        <!--optional, whether the remaining door closed is not supported-->
    </isNotSupportNormalClose>
    <isNotSupportAlarmout>
        <!--optional, whether the alarm output linkage is not supported-->
    </isNotSupportAlarmout>
    <isNotSupportCapturePic>
        <!--optional, whether the capture linkage is not supported-->
    </isNotSupportCapturePic>
    <isNotSupportMainDevBuzzer>
        <!--optional, whether not supports buzzing linkage of access controller,

```

```

if supports, this node will not return-->
</isNotSupportMainDevBuzzer>
<isNotSupportReaderBuzzer>
    <!--optional, whether not supports buzzing linkage of card reader, if
supports, this node will not returned-->
    </isNotSupportReaderBuzzer>
    <purePwdVerifyEnable><!--optional, boolean, whether the device supports
opening the door only by password: true-yes, this node is not returned-no--></
purePwdVerifyEnable>
    <!--For opening the door only by password: 1. The password in "XXX or
password" in the authentication mode refers to the person's password (the value
of the node password in JSON_UserInfo); 2. The device will not check the
duplication of the password, and the upper platform should ensure that the
password is unique; 3. The password cannot be added, deleted, edited, or
searched for on the device locally-->
    <EventList>
        <EventEntry>
            <Index>0</Index>
            <mainEventName>DevEvent</mainEventName>
            <SubEventNameList>
                <subEventName>hostAntiDismantle</subEventName>
                <!--required, controller tampering alarm -->
                <subEventName>OfflineEcentNearlyFull</subEventName>
                <!--required, alarm for offline event exceeding 90% -->
                <subEventName>NetBroken</subEventName>
                <!--required, network disconnected -->
                <subEventName>NetRume</subEventName>
                <!--required, network recovery -->
                <subEventName>LowBattery</subEventName>
                <!--required, battery low voltage -->
                <subEventName>BatteryReume</subEventName>
                <!--required, battery voltage recovered -->
                <subEventName>ACOff</subEventName>
                <!--required, AC power off -->
                <subEventName>ACResume</subEventName>
                <!--required, AC power recovery-->
                <subEventName>SDCardFull</subEventName>
                <!--required, SD card full alarm-->
                <subEventName>LinkageCapturePic</subEventName>
                <!--required, Linked capture event alarm-->
                <subEventName>ImageQualityLow</subEventName>
                <!--required, low face picture quality-->
                <subEventName>FingerPrintQualityLow</subEventName>
                <!--required, low fingerprint picture quality-->
                <subEventName>BatteryElectricLow</subEventName>
                <!--required, low battery voltage (for face device only)-->
                <subEventName>BatteryElectricResume</subEventName>
                <!--required, battery voltage recovery (for face device only)-->
                <subEventName>FireImportShortCircuit</subEventName>
                <!--req fire input short-circuit alarm-->
                <subEventName>FireImportBrokenCircuit</subEventName>
                <!--req fire input broken-circuit alarm-->

```

```

<subEventName>FireImportResume</subEventName>
<!--req fire input recovery-->
<subEventName>MasterRS485LoopnodeBroken</subEventName>
<!--req main controller RS485 loop node disconnection-->
<subEventName>MasterRS485LoopnodeResume</subEventName>
<!--req main controller RS485 loop node connection recovery-->
<subEventName>DistractControllerOnLine</subEventName>
<!--required, Distributed controller online-->
<subEventName>DistractControllerOffLine</subEventName>
<!--required, Distributed controller offline-->
<subEventName>FireButtonTrigger</subEventName>
<!--required, Fire button triggered-->
<subEventName>FireButtonResume</subEventName>
<!--required, Fire button recovered-->
<subEventName>MaintenanceButtonTrigger</subEventName>
<!--required, Maintenance button triggered-->
<subEventName>MaintenanceButtonResume</subEventName>
<!--required, Maintenance button recovered-->
<subEventName>EmergencyButtonTrigger</subEventName>
<!--required, Emergency button triggered-->
<subEventName>EmergencyButtonResume</subEventName>
<!--required, Emergency button recovered-->
<subEventName>LocalControlOffline</subEventName>
<!--req distributed access controller offline-->
<subEventName>LocalControlResume</subEventName>
<!--required, distributed access controller connection recovered-->
<subEventName>LocalDownsideRS485LoopNodeBroken</subEventName>
<!--required, distributed access controller downlink RS485 loop
disconnection-->
<subEventName>LocalDownsideRS485LoopNodeResume</subEventName>
<!--required, distributed access controller downlink RS485 loop
connection recovered-->
<subEventName>SubmarinebackCommBreak</subEventName>
<!--required, disconnected with anti-passing back server-->
<subEventName>SubmarinebackCommResume</subEventName>
<!--required, resume connection with anti-passing back server-->
<subEventName>RemoteActualGuard</subEventName>
<!--required, remote real-time arming-->
<subEventName>RemoteActualUnguard</subEventName>
<!--required, remote real-time disarming-->
<subEventName>MotorSensorException</subEventName>
<!--required, motor or sensor exception-->
<subEventName>CanBusException</subEventName>
<!--required, CAN bus exception-->
<subEventName>CanBusResume</subEventName>
<!--required, CAN bus restored-->
<subEventName>GateTemperatureOverrun</subEventName>
<!--required, too high pedestal temperature-->
<subEventName>IREmitterException</subEventName>
<!--required, active infrared intrusion detector exception-->
<subEventName>IREmitterResume</subEventName>
<!--required, active infrared intrusion detector restorted-->

```

```

<subEventName>LampBoardCommException</subEventName>
<!--required, communication with light board failed-->
<subEventName>LampBoardCommResume</subEventName>
<!--required, communication with light board restored-->
<subEventName>IRAdaptorBoardCommException</subEventName>
<!--required, communicated with IR adaptor exception-->
<subEventName>IRAdaptorBoardCommResume</subEventName>
<!--required, communication with IR adaptor restored-->
<subEventName>ChannelControllerDesmantleAlarm</subEventName>
<!--required, lane controller tampering alarm-->
<subEventName>ChannelControllerDesmantleResume</subEventName>
<!--required, lane controller tampering alarm restored-->
<subEventName>ChannelControllerFireImportAlarm</subEventName>
<!--required, lane controller fire input alarm-->
<subEventName>ChannelControllerFireImportResume</subEventName>
<!--required, lane controller fire input alarm restored-->
<subEventName>StayEvent</subEventName>
<!--optional, loitering event-->
<subEventName>LegalEventNearlyFull</subEventName>
<!--optional, alarm of no memory for legal offline event storage-->
</SubEventNameList>
</EventEntry>
<EventEntry>
    <Index>1</Index>
    <mainEventName>AlarmEvent</mainEventName>
    <SubEventNameList>
        <subEventName>AlarminShortCircuit</subEventName>
        <!--required, zone short circuit alarm-->
        <subEventName>AlarminBrokenCircuit</subEventName>
        <!--required, zone open circuit alarm -->
        <subEventName>AlarminException</subEventName>
        <!--required, zone exception alarm -->
        <subEventName>AlarmResume</subEventName>
        <!--required, zone alarm recovery -->
        <subEventName>CaseSensorAlarm</subEventName>
        <!--required, event input alarm -->
        <subEventName>CaseSensorResume</subEventName>
        <!--required, event input recovery -->
    </SubEventNameList>
</EventEntry>
<EventEntry>
    <Index>2</Index>
    <mainEventName>DoorEvent</mainEventName>
    <SubEventNameList>
        <subEventName>LeaderCardOpenBegin</subEventName>
        <!--required, first card opening door starts -->
        <subEventName>LeaderCardOpenStop</subEventName>
        <!--required, first card open status door ends -->
        <subEventName>AlwaysOpenBegin</subEventName>
        <!--required, remained open status starts -->
        <subEventName>AlwaysOpenStop</subEventName>
        <!--required, remained open status ends -->
    </SubEventNameList>
</EventEntry>

```

```

<subEventName>AlwaysCloseBegin</subEventName>
<!--required, remained closed status ends -->
<subEventName>AlwaysCloseStop</subEventName>
<!--required, remaining closed status ends-->
<subEventName>LockOpen</subEventName>
<!--required, open the door lock -->
<subEventName>LockClose</subEventName>
<!--required, close the lock -->
<subEventName>DoorButtonPress</subEventName>
<!--required, exit button pressed -->
<subEventName>DoorButtonRelease</subEventName>
<!--required, exit button released -->
<subEventName>DoorOpenNormal</subEventName>
<!--required, normally open the door (door magnetic) -->
<subEventName>DoorCloseNormal</subEventName>
<!--required, normally close the door (door magnetic) -->
<subEventName>DoorOpenAbnormal</subEventName>
<!--required, door opening exception (door magnetic )-->
<subEventName>DoorOpenTimeout</subEventName>
<!--required, door opening timeout (door magnetic )-->
<subEventName>RemoteOpenDoor</subEventName>
<!--required, remotely open the door-->
<subEventName>RemoteCloseDoor</subEventName>
<!--required, remotely closed the door-->
<subEventName>RemoteAlwaysOpen</subEventName>
<!--required, remotely remain open -->
<subEventName>RemoteAlwaysClose</subEventName>
<!--required, remotely remain closed -->
<subEventName>NotBelongMultiCard</subEventName>
<!--required, the card is not associated to the multi-authentication
group-->
<subEventName>InvalidMultiVerifyPeriod</subEventName>
<!--required, the card is not in the multi-authentication time period
-->
<subEventName>MultiVerifySuperRightFail</subEventName>
<!--required, super password authentication failed -->
<subEventName>MultiVerifyRemoteRightFail</subEventName>
<!--required, remote authentication failed -->
<subEventName>MultiVerifySuccess</subEventName>
<!--required, successfully multi -authentication -->
<subEventName>MultiVerifyNeedRemoteOpen</subEventName>
<!--required, multi-authentication needs remote opening door -->
<subEventName>MultiVerifySuperRightSuccess</subEventName>
<!--required, successfully super password -->
<subEventName>MultiVerifyRepeatFail</subEventName>
<!--required, repeat authentication failed -->
<subEventName>MultiVerifyTimeout</subEventName>
<!--required, multi-authentication timeout -->
<subEventName>RemoteCapturePic</subEventName>
<!--required,remote capture-->
<subEventName>DoorBellRing</subEventName>
<!--required,door bell ringing-->

```

```
<subEventName>CallCenter</subEventName>
<!--required, call center-->
<subEventName>FirstCardAuthorizeBegin</subEventName>
<!--required, first card authorization started-->
<subEventName>FirstCardAuthorizeEnd</subEventName>
<!--required, first card authorization ended-->
<subEventName>FirstCardOpenWithoutAuthorize</subEventName>
<!--required, open door with unauthorized first card failed.-->
<subEventName>SecurityMoudleDesmantleAlarm</subEventName>
<!--required, door control security module anti-tamper alarm-->
<subEventName>FirstCardAuthorizeBegin</subEventName>
<!--req first card authorization start-->
<subEventName>FirstCardAuthorizeEnd</subEventName>
<!--req first card authorization end-->
<subEventName>DoorLockInputShortCircuit</subEventName>
<!--req door lock input short-circuit alarm-->
<subEventName>DoorLockInputBrokenCircuit</subEventName>
<!--req door lock input broken-circuit alarm-->
<subEventName>DoorLockInputException</subEventName>
<!--req door lock input exception alarm-->
<subEventName>DoorContactInputShortCircuit</subEventName>
<!--req magnet input short-circuit alarm-->
<subEventName>DoorContactInputBrokenCircuit</subEventName>
<!--req magnet input broken-circuit alarm-->
<subEventName>DoorContactInputException</subEventName>
<!--req magnet input exception alarm-->
<subEventName>OpenButtonInputShortCircuit</subEventName>
<!--req door button input short-circuit alarm-->
<subEventName>OpenButtonInputBrokenCircuit</subEventName>
<!--req door button input broken-circuit alarm-->
<subEventName>OpenButtonInputException</subEventName>
<!--req door button input exception alarm-->
<subEventName>DoorLockOpenException</subEventName>
<!--req door lock open exception-->
<subEventName>DoorLockOpenTimeout</subEventName>
<!--req door lock open timeout-->
<subEventName>FirstCardOpenWithoutAuthorize</subEventName>
<!--req first card failed to open door without authorization-->
<subEventName>CallLadderRelayBreak</subEventName>
<!--required,Elevator relay disconnected-->
<subEventName>CallLadderRelayClose</subEventName>
<!--required,Elevator relay connected-->
<subEventName>AutoKeyRelayBreak</subEventName>
<!--required,Auto-button relay disconnected-->
<subEventName>AutoKeyRelayClose</subEventName>
<!--required,Auto-button relay connected-->
<subEventName>KeyControlRelayBreak</subEventName>
<!--required,Button relay disconnected-->
<subEventName>KeyControlRelayClose</subEventName>
<!--required,Button relay connected-->
<subEventName>RemoteVisitorCallLadder</subEventName>
<!--required,Visitor called elevator-->
```

```

<subEventName>RemoteHouseholdCallLadder</subEventName>
<!--required, Resident called elevator-->
<subEventName>LegalMessage</subEventName>
<!--required, valid message-->
<subEventName>IllegalMessage</subEventName>
<!--required, invalid message-->
<subEventName>Trailing</subEventName>
<!--required, tailgating-->
<subEventName>ReverseAccess</subEventName>
<!--required, reserve passing-->
<subEventName>ForceAccess</subEventName>
<!--required, force accessing-->
<subEventName>ClimbingOverGate</subEventName>
<!--required, climbing over barrier-->
<subEventName>PassingTimeout</subEventName>
<!--required, passing timed out-->
<subEventName>IntrusionAlarm</subEventName>
<!--required, intrusion alarm-->
<subEventName>FreeGatePassNotAuth</subEventName>
<!--required, authentication failed when free passing the turnstile-->
<subEventName>DropArmBlock</subEventName>
<!--required, barrier obstructed-->
<subEventName>DropArmBlockResume</subEventName>
<!--required, barrier obstruction restored-->
<subEventName>RemoteControlCloseDoor</subEventName>
<!--required, close door via keyfob-->
<subEventName>RemoteControlOpenDoor</subEventName>
<!--required, open door via keyfob-->
<subEventName>RemoteControlAlwaysOpenDoor</subEventName>
<!--required, remain door open via keyfob-->
</SubEventNameList>
</EventEntry>
<EventEntry>
<Index>3</Index>
<mainEventName>ReaderEvent</mainEventName>
<SubEventNameList>
<subEventName>StressAlarm</subEventName>
<!--required, duress alarm-->
<subEventName>ReaderDesmantleAlarm</subEventName>
<!--required, card reader tamper-proof alarm-->
<subEventName>LegalCardPass</subEventName>
<!--required, valid card successfully authenticated -->
<subEventName>CardAndPasswdPass</subEventName>
<!--required, card and password successfully authenticated -->
<subEventName>CardAndPasswdFail</subEventName>
<!--required, card and password authentication failed -->
<subEventName>CardAndPasswdTimeout</subEventName>
<!--required, card and password authentication timeout -->
<subEventName>CardMaxAuthenticateFail</subEventName>
<!--required, card reader authentication over times -->
<subEventName>CardNoRight</subEventName>

```

```
<!--required, no permission for the card -->
<subEventName>CardInvalidPeriod</subEventName>
<!--required, invalid time segment -->
<subEventName>CardOutOfDate</subEventName>
<!--required, card exceeds the validate -->
<subEventName>InvalidCard</subEventName>
<!--required, invalid card No. -->
<subEventName>AntiSneakFail</subEventName>
<!--required, anti-passback authentication failed -->
<subEventName>InterlockDoorNotClose</subEventName>
<!--required, interlocking door not closed -->
<subEventName>FingerprintComparePass</subEventName>
<!--required, Fingerprint Recognition Passed-->
<subEventName>FingerprintCompareFail</subEventName>
<!--required, Fingerprint Recognition Failed-->
<subEventName>CardFingerprintVerifyPass</subEventName>
<!--required, Card + Fingerprint Authentication Passed-->
<subEventName>CardFingerprintVerifyFail</subEventName>
<!--required, Card + Fingerprint Authentication Failed-->
<subEventName>CardFingerprintVerifyTimeout</subEventName>
<!--required, Card + Fingerprint Authentication Timeout-->
<subEventName>CardFingerprintPasswdVerifyPass</subEventName>
<!--required, Card + Fingerprint + Password Authentication Passed-->
<subEventName>CardFingerprintPasswdVerifyFail</subEventName>
<!--required, Card + Fingerprint + Password Authentication Failed-->
<subEventName>CardFingerprintPasswdVerifyTimeout</subEventName>
<!--required, Card + Fingerprint + Password Authentication Timeout-->
<subEventName>FingerprintPasswdVerifyPass</subEventName>
<!--required, Fingerprint + Password Authentication Passed-->
<subEventName>FingerprintPasswdVerifyFail</subEventName>
<!--required, Fingerprint + Password Authentication Failed-->
<subEventName>FingerprintPasswdVerifyTimeout</subEventName>
<!--required, Fingerprint + Password Authentication Timeout-->
<subEventName>FingerprintInexistence</subEventName>
<!--required, No Fingerprint-->
<subEventName>FaceVerifyPass</subEventName>
<!--required, Face Authentication Passed-->
<subEventName>FaceVerifyFail</subEventName>
<!--required, Face Authentication Failed-->
<subEventName>FaceAndFpVerifyPass</subEventName>
<!--required, Face + Fingerprint Authentication Passed-->
<subEventName>FaceAndFpVerifyFail</subEventName>
<!--required, Face + Fingerprint Authentication Failed-->
<subEventName>FaceAndFpVerifyTimeout</subEventName>
<!--required, Face + Fingerprint Authentication Timeout-->
<subEventName>FaceAndPwVerifyPass</subEventName>
<!--required, Face + Password Authentication Passed-->
<subEventName>FaceAndPwVerifyFail</subEventName>
<!--required, Face + Password Authentication Failed-->
<subEventName>FaceAndPwVerifyTimeout</subEventName>
<!--required, Face + Password Authentication Timeout-->
<subEventName>FaceAndCardVerifyPass</subEventName>
```

```

<!--required, Face + Card Authentication Passed-->
<subEventName>FaceAndCardVerifyFail</subEventName>
<!--required, Face + Card Authentication Failed-->
<subEventName>FaceAndCardVerifyTimeout</subEventName>
<!--required, Face + Card Authentication Timeout-->
<subEventName>FaceAndPwAndFpVerifyPass</subEventName>
<!--required, Face + Password + Fingerprint Authentication Passed-->
<subEventName>FaceAndPwAndFpVerifyFail</subEventName>
<!--required, Face + Password + Fingerprint Authentication Failed-->
<subEventName>FaceAndPwAndFpVerifyTimeout</subEventName>
<!--required, Face + Password + Fingerprint Authentication Timeout-->
<subEventName>FaceAndCardAndFpVerifyPass</subEventName>
<!--required, Face + Card + Fingerprint Authentication Passed-->
<subEventName>FaceAndCardAndFpVerifyFail</subEventName>
<!--required, Face + Card + Fingerprint Authentication Failed-->
<subEventName>FaceAndCardAndFpVerifyTimeout</subEventName>
<!--required, Face + Card + Fingerprint Authentication Timeout-->
<subEventName>EmployeeAndFpVerifyPass</subEventName>
<!--required, Employee No. + Fingerprint Authentication Passed-->
<subEventName>EmployeeAndFpVerifyFail</subEventName>
<!--required, Employee No. + Fingerprint Authentication Failed-->
<subEventName>EmployeeAndFpVerifyTimeout</subEventName>
<!--required, Employee No. + Fingerprint Authentication Timeout-->
<subEventName>EmployeeAndFpAndPwVerifyPass</subEventName>
<!--required, Employee No. + Fingerprint + Password Authentication
Passed-->
<subEventName>EmployeeAndFpAndPwVerifyFail</subEventName>
<!--required, Employee No. + Fingerprint + Password Authentication
Failed-->
<subEventName>EmployeeAndFpAndPwVerifyTimeout</subEventName>
<!--required, Employee No. + Fingerprint + Password Authentication
Timeout-->
<subEventName>EmployeeAndFaceVerifyPass</subEventName>
<!--required, Employee No. + Face Authentication Passed-->
<subEventName>EmployeeAndFaceVerifyFail</subEventName>
<!--required, Employee No. + Face Authentication Failed-->
<subEventName>EmployeeAndFaceVerifyTimeout</subEventName>
<!--required, Employee No. + Face Authentication Timeout-->
<subEventName>FaceRecognizeFail</subEventName>
<!--required, Face picture recognition failed-->
<subEventName>EmployeeAndPwVerifyPass</subEventName>
<!--required, Employee No. + Password Authentication Passed-->
<subEventName>EmployeeAndPwVerifyFail</subEventName>
<!--required, Employee No. + Password Authentication Failed-->
<subEventName>EmployeeAndPwVerifyTimeout</subEventName>
<!--required, Employee No. + Password Authentication Timeout-->
<subEventName>DoorOpenOrDormantFail</subEventName>
<!--required, door remains closed or sleepy status authentication
failed.-->
<subEventName>AuthPlanDormantFail</subEventName>
<!--required, authentication of sleepy mode in the schedule failed.-->
<subEventName>CardEncryptVerifyFail</subEventName>

```

```

<!--required, authentication of card encryption failed.-->
<subEventName>SubmarinebackReplyFail</subEventName>
<!--required, response of anti-passing back server failed.-->
<subEventName>PasswordMismatch</subEventName>
<!--optional, password mismatched.-->
<subEventName>EmployeeNoNotExist</subEventName>
<!--required, the employee ID does not exist.-->
<subEventName>CombinedVerifyPass</subEventName>
<!--required, authenticated .-->
<subEventName>CombinedVerifyTimeout</subEventName>
<!--required, authentication timed out.-->
<subEventName>VerifyModeMismatch</subEventName>
<!--required, authentication mode mismatched.-->
<subEventName>PasswordVerifyPass</subEventName>
<!--optional, password authenticated-->
<subEventName>HumanDetectFail</subEventName>
<!--required, human detection failed.-->
<subEventName>PeopleAndIdCardComparePass</subEventName>
<!--required, face and ID card authenticated-->
<subEventName>PeopleAndIdCardCompareFail</subEventName>
<!--required, face and ID card authentication failed-->
<subEventName>InformalMifareCardVerifyFail</subEventName>
<!--optional, authentication failed: invalid Mifare card-->
<subEventName>CPUCardEncryptVerifyFail</subEventName>
<!--optional, verifying CPU card encryption failed-->
<subEventName>NFCDisableVerifyFail</subEventName>
<!--optional, disabling NFC verification failed-->
<subEventName>EMCardRecognizeNotEnabled</subEventName>
<!--optional, EM card recognition is disabled-->
<subEventName>M1CardRecognizeNotEnabled</subEventName>
<!--optional, M1 card recognition is disabled-->
<subEventName>CPUCardRecognizeNotEnabled</subEventName>
<!--optional, CPU card recognition is disabled-->
<subEventName>IDCardRecognizeNotEnabled</subEventName>
<!--optional, ID card recognition is disabled-->
<subEventName>CardSetSecretKeyFail</subEventName>
  <!--optional, importing key to the card failed-->
</SubEventNameList>
</EventEntry>
</EventList>
</EventLinkage>

<FingerPrint>
  <!--required, fingerprint parameters -->
  <enable opt="true,false"/>
  <!--required, whether to support fingerprint settings -->
  <cardNo min="" max="" />
  <!--required, card No. length -->
  <fingerPrintLen min="" max="" />
  <!--required, fingerprint data length-->
  <EnableCardReader min="" max="" />
  <!--required, supported card reader No.-->

```

```

<fingerType opt="Normal,Stress,patrolFP,superFP,dissmissingFP"/>
  <!--required, "Normal"-normal fingerprint, "Stress"-duress fingerprint,
  "patrolFP"-patrol fingerprint, "superFP"-super fingerprint, "dissmissingFP"-  

  dismiss fingerprint-->
    <fingerPrintID min="" max="" />
      <!--required, finger ID-->
      <callbackMode opt="allRetrun,partReturn"/>
        <!--required, callback mode, allRetrun-block (return after all the card  

        readers are offline), partReturn-non-block (return after a part of card readers  

        are offline)-->
        <isSupportFingerNo/>
          <!--optional, boolean, whether the device supports setting finger ID:  

          "true"-yes-->
            <recvStatus opt="0,1,2,3,4,5,6,7,8,9,10"/>
              <!--optional, error status: 0-success, 1-incorrect finger ID, 2-incorrect  

              fingerprint type, 3-invalid card No. (the card No. does not meet the device  

              requirements), 4-the fingerprint is not linked with employee No. or card No.  

              (the employee No. or the card No. is NULL), 5-the employee No. does not exist,  

              6-the fingerprint data length is 0, 7-invalid card reader No., 8-invalid  

              employee No., 9-invalid first-time authentication value, 10-other parameters  

              error-->
                <employeeNo min="" max="" />
                  <!--optional, employee No. (person ID)-->
                  <leaderFP opt="true"/>
                    <!--optional, whether the fingerprint supports first-time authentication:  

                    "true"-yes, "false" or this node is not returned-no-->
                      <isSupportFingerCover>
                        <!--optional, xs:boolean, whether to overwrite the old fingerprint  

                        information when applying a new fingerprint information linked to the same  

                        employee No. (person ID): "true"-yes, this node is not returned-no-->
                      </isSupportFingerCover>
                    </FingerPrint>
                    <DelFingerPrint>
                      <!--required, delete fingerprint parameter, which corresponds to the  

                      command NET_DVR_DEL_FINGERPRINT_CFG_V50. This node will not be returned if  

                      device does not support this function. After calling the API  

                      NET_DVR_StartRemoteConfig with command NET_DVR_DEL_FINGERPRINT_CFG_V50, if this  

                      node is returned, you should wait for the return of callback function to get  

                      the actual deleting result; if this node is not returned, the return of API  

                      NET_DVR_StartRemoteConfig already indicates the deleting result-->
                      <delFingerPrintMode opt="byCard,byReader"/>
                        <!--required, deleting fingerprint mode: byCard-by card No., byReader- by  

                        card reader-->
                      <FingerPrintStatus>
                        <!--required, delete fingerprint status-->
                        <cardReaderNo min="" max="" />
                          <!--required, fingerprint recorder No.-->
                          <status min="0" max="3"/>
                            <!--required, status: 0-invalid, 1-handling, 2-deleting failed, 3-  

                            completed-->
                          </FingerPrintStatus>
                        <employeeNo min="" max="" />

```

```

<!--required, employee No. (person ID)-->
</DelFingerPrint>

<SMS>
<enable opt="true,false"/>
<!--required, whether to support SMS funtion -->
<PhoneLinkageDoor>
    <!--required, mobile phone links with door -->
    <openRight opt="true,false"/>
    <!--required, door opening permission -->
    <closeRight opt="true,false"/>
    <!--required, door closing permission -->
    <NormalOpenRight opt="true,false"/>
    <!--required, door remained opening permission -->
    <NormalCloseRight opt="true,false"/>
    <!--required, door remained closing permission -->
    <armRight opt="true,false"/>
    <!--required, arming permission -->
    <DisarmRight opt="true,false"/>
    <!--required, disarming permission -->
</PhoneLinkageDoor>
<whiteListNum min="1" max="32"/>
<!--required, allowlist number-->
</SMS>
<RealteUserInfo>
    <!--required, NET_DVR_CARD_CFG_SEND_DATA and
NET_DVR_CARD_USER_INFO_CFG-->
    <enabled opt="true,false"/>
    <!--required, whether to support card No. being linked to user
information-->
    <userNameLen min="" max="" />
    <!--required, user name length-->
</RealteUserInfo>
<ContinuousShootCfg>
    <!--required,NET_DVR_SNAPCFG-->
    <enabled opt="true,false"/>
    <!--required,whether to support triggering capture parameters
configuration-->
    <relatedDriveWay min="" max="" />
    <!--required, IO related vehicle lane No.-->
    <snapTimes min="" max="" />
    <!--required, coil capture times:, 0-5-->
    <snapWaitTime min="" max="" />
    <!--required, capture waiting time, unit:ms, value range[0,60000]-->
    <IntervalTimeList size="4">
        <intervalTime min="" max="" />
        <!--required,interval of continuous capture, unit:ms-->
    </IntervalTimeList>
    <JpegParam>
        <picSize
opt="CIF,QCIF,D1,UXGA,SVGA,HD720P,VGA,XVGA,HD900p,HD1080,2560*1920,1600*304,2048
*1536,2448*2048,2448*1200,

```

```
2448*800,XGA,SXGA,WD1,1080i,  
576*576,1536*1536,1920*1920,320*240,720*720,1024*768,1280*1280,1600*600,  
2048*768,160*120,336*256,384*256,384*216,320*256,320*192,512*384,480*272,512*272  
,288*320,144*176,  
480*640,240*320,120*160,576*720,720*1280,576*960, 180*240,  
360*480, 540*720, 720*960, 960*1280, 1080*1440, Auto"/>  
    <!-- optional,image size-->  
    <picQuality opt="best,good,general" />  
    <!-- optional,image quality: 0-Best, 1- Better, 2- Good-->  
</JpegParam>  
</ContinuousShootCfg>  
<PictureCfg>  
    <!--required,reuse some fields of NET_DVR_PICTURECFG-->  
    <enableUp opt="true,false"/>  
    <!--required, whether to support background picture uploading-->  
    <enableDel opt="true,false"/>  
    <!--required, whether to support deleting background picture-->  
    <useType min="" max="" />  
    <!--required,picture type, 1- background picture, 2-GIF picture, 3-CAD  
picture-->  
    <sequence min="" max="" />  
    <!--required, sequence No.-->  
    <BasemapCfg>  
        <sourWidth min="" max="" />  
        <!--required, initial picture width-->  
        <sourHeight min="" max="" />  
        <!--required, initial picture height-->  
    </BasemapCfg>  
</PictureCfg>  
<ExternalDevCfg>  
  
    <!--required,NET_DVR_ACS_EXTERNAL_DEV_CFG-->  
    <IDCardUpMode opt="number,all"/>  
    <!--required, ID information report, number: upload 18-digit ID number;  
all: upload all information-->  
    <cardVerifyMode opt="remoteCenter,clientPlatform"/>  
    <!--required, card verification mode, remoteCenter: remote center  
verification; clientPlatform: client platform verification-->  
    <ACSDevType  
opt="IDCardReader,ICReader,QRCodeReader,fingerPrintReader,QRCodeReaderandScreen,  
recycleCard,screen,fingerPrintModule,voiceModule,peopleAndIdCard"/>  
    <!--required, device model: 1- ID card reader, 2- IC card reader, 3- QR  
code reader, 4- Fingerprint reader, 5- Screen + QR code reader, 6- Card  
collector, 7- Screen, 8- Fingerprint scanner, 9- Voice module, 10-person and ID  
card device-->  
    <doorMode opt="inDoor,outDoor"/>  
    <!--required, door in/out type, inDoor: enter, outDoor: exit-->  
    <DevDetailType>  
        <IDCardReaderType opt="iDR210, IDM10, HikIDCardReader"/>  
        <!--required, ID card reader model-->  
        <screenType opt="DC48270RS043_01T,DC80480B070_03T"/>
```

```

    <!--required, LCD model-->
</DevDetailType>
</ExternalDevCfg>
<PersonnelChannelCfg>
    <!--required,NET_DVR_PERSONNEL_CHANNEL_CFG-->
    <inMode opt="controlled,forbid,freedom"/>
    <!--required, enter mode, 0- controlled; 1- denied; 2- free-->
    <outMode opt="controlled,forbid,freedom"/>
    <!--required, exit mode, 0- controlled; 1- denied; 2- free-->
    <workMode opt="urgent,repair,normalClose,normalOpen"/>
    <!--required, operating mode, 0- emergency, 1- maintenance, 2- normally
closed, 3- normally open-->
</PersonnelChannelCfg>
<PlatformVerifyCfg>
    <!--required,NET_DVR_PLATFORM_VERIFY_CFG-->
    <doorNo min="" max="" />
    <!--required, door No.-->
    <resultType opt="legal,illegal"/>
    <!--required, verification result type, legal: illegal, illegal: legal-->
    <screenDisplay min="" max="" />
    <!--required,LED display character length-->
</PlatformVerifyCfg>
<PersonStatisticsCfg>
    <!--required,NET_DVR_PERSON_STATISTICS_CFG-->
    <enableStatistics opt="true,false"/>
    <!--required, whether to enable people counting-->
    <enableOfflineStatistics opt="true,false"/>
    <!--required, whether to enable offline people counting-->
    <countSignalStatisticalStandard opt="IRDetectPass,AuthQuantity"/>
    <!--required, people counting type: IRDetectPass- by IR detection,
AuthQuantity- by authentication number-->
</PersonStatisticsCfg>
<ScreenDisplayCfg>
    <!--required,NET_DVR_SCREEN_DISPLAY_CFG-->
    <FontSize min="" max="" />
    <!--required, font size-->
    <rowSpacing min="" max="" />
    <!--required, row space-->
    <columnSpacing min="" max="" />
    <!--required, column space-->
    <firstRowPosition opt="0,1/8,2/8,3/8,4/8,5/8,6/8,7/8"/>
    <!--required, first row position-->
    <degree opt="0,90"/>
    <!--required, character display direction abgle, unit: degree-->
    <screenType opt="DC48270RS043_01T,DC80480B070_03T"/>
    <!--required, screen type-->
</ScreenDisplayCfg>
<GateTimeCfg>
    <!--required,NET_DVR_GATE_TIME_CFG-->
    <holdOnALarmTime min="" max="" />
    <!--required, extend alarm buzzer time, unit: ms -->
    <holdOnGateOpenTime min="" max="" />

```

```

<!--required, door open time before receiving close command, unit: ms-->
<postponeIntrusionAlarmTime min="" max="" />
<!--required, delay trigger intrusion alarm time, unit: ms-->
<noLaneAccessTimeLimitTime min="" max="" />
<!--required, timeout alarm time for no people passing after channel
received valid passing verification signal, unit: s-->
<safetyZoneStayTime min="" max="" />
<!--required, timeout alarm time for people staying in the channel when
reached safety region after the channel received valid passing verification
signal, unit:s-->
<IRTriggerTimeoutTime min="0" max="255"/>
<!--required, IR triggering timeout, unit: s-->
</GateTimeCfg>
<LocalControllerStatus>enable</LocalControllerStatus>
<!--required, support getting distributed access controller status-->
<searchLocalController>enable</searchLocalController>
<!--required, support searching distributed access controller-->
<showDeviceType opt="Floor"/>
<!--optional,Display device type (by default, display the door parameters if
there is no this field),Floor- Displayed floor-->

<FaceParam>
    <!--required,Face parameter-->
    <enable opt="true"/>
    <!--required,whether to support face parameter configuration-->
    <cardNo min="" max="" />
    <!--required,Card No. length-->
    <faceLen min="" max="" />
    <!--required,Face data length-->
    <enableCardReader min="" max="" />
    <!--required,Supported card reader No.-->
    <faceID min="" max="" />
    <!--required,Face No.-->
    <faceDataType opt="module,picture"/>
    <!--required,Face data type (the default type is template if there is no
this node)-->
    <isSupportFaceCover>
        <!--optional, whether supports covering existed data when applying face
picture data-->
    </isSupportFaceCover>
</FaceParam>
<isSupportGetDeviceEvent opt="true,false"/>
<!--optional, whether to support getting device event: "true"-yes, "false" or
this node is not returned-no-->
<isSupportDeployType min="0" max="1"/>
    <!--optional, supported arming type: 0-arm via client software, 1-real-time
arming-->
<UploadRightControllerAudio>
    <!--required, uploading audio file of main controller-->
    <audioID min="2" max="32"/>
        <!--required, audio file ID. 0xffffffff indicates uploading all audio
files, and currently the device only supports uploading all audio files instead

```

```

of uploading a single audio file by ID-->
</UploadRightControllerAudio>
<DownloadRightControllerAudio>
    <!--required, downloading audio file of main controller-->
    <audioID min="2" max="32"/>
        <!--required, audio file ID. 0xffffffff indicates downloading all audio
files, and currently the device only supports downloading all audio files
instead of downloading a single audio file by ID-->
    </DownloadRightControllerAudio>
    <BlackListPictureParam>
        <!--required, parameter of picture in blocklist
(NET_DVR_BULK_UPLOAD_BLOCKLIST_PICTURE)-->
        <BlackListPictureCond>
            <!--required, blocklist picture condition-->
            <pictureNum min="" max="" />
                <!--required, picture quantity-->
        </BlackListPictureCond>
        <cardNo min="" max="" />
            <!--required, card No.-->
        <name min="" max="" />
            <!--required, name-->
        <sex opt="male,female"/>
            <!--required, gender: male- Male, female- Female-->
        <pictureValid opt="invalid,valid"/>
            <!--required, whether blocklist picture is valid: invalid- Invalid,valid?
Valid-->
            <pictureLen min="" max="" />
                <!--required, blocklist picture size-->
        <BlackListPictureStatus>
            <!--required, blocklist picture status-->
            <cardNo min="" max="" />
                <!--required, card No.-->
            <status opt=" processing,failed,success"/>
                <!--required, status: processing- Processing, failed- Failed,success-
Succeeded-->
            </BlackListPictureStatus>
        </BlackListPictureParam>
        <IDBlackListParam>
            <!--ID blocklist parameter (NET_DVR_BULK_UPLOAD_ID_BLOCKLIST)-->
            <IDBlackListCond>
                <!--required, ID blocklist condition-->
                <blackListNum min="" max="" />
                    <!--required, blocklist quantity-->
            </IDBlackListCond>
            <blackListValid opt="invalid,valid"/>
            <!--required, whether ID card blocklist is valid or not-->
            <IDBlackListStatus>
                <!--required, ID card blocklist status-->
                <IDNum min="" max="" />
                    <!--required, ID number-->
                <status opt=" processing,failed,success"/>
                <!--required, status: processing- Processing, failed- Failed, success-

```

```

Succeeded-->
    </IDBlackListStatus>
</IDBlackListParam>
<CaptureFingerPrint>
    <!--optional, xs:boolean, collect fingerprint information-->
    <pictureType opt="full,quarter">
        <!--required, xs:string, fingerprint picture type-->
    </pictureType>
    <fingerNo min="1" max="10">
        <!--required, xs:integer, fingerprint No.-->
    </fingerNo>
    <isSupportFingerData opt="true,false">
        <!--required, xs:boolean, fingerprint data-->
    </isSupportFingerData>
    <isSupportFingerPicture opt="true,false">
        <!--required, xs:boolean, fingerprint picture-->
    </isSupportFingerPicture>
    <fingerPrintQuality min="1" max="100">
        <!--required, xs:integer, fingerprint quality-->
    </fingerPrintQuality>
</CaptureFingerPrint>
<CaptureFace>
    <!--optional, xs:boolean, collect face information-->
    <isSupportFaceTemplate1 opt="true,false">
        <!--required, xs:boolean, face template data 1-->
    </isSupportFaceTemplate1>
    <isSupportFaceTemplate2 opt="true,false">
        <!--required, xs:boolean, face template data 2-->
    </isSupportFaceTemplate2>
    <isSupportFacePic opt="true,false">
        <!--required, xs:boolean, face picture data-->
    </isSupportFacePic>
    <faceQuality min="1" max="100">
        <!--required, xs:integer, face quality-->
    </faceQuality>
    <captureProgress opt="0,100">
        <!--required, xs:integer, collection progress-->
    </captureProgress>
    <isSupportInfraredFacePic opt="true,false"><!--required, xs:boolean, whether
to support infrared face picture data--></isSupportInfraredFacePic>
</CaptureFace>
<isSupportUploadCertificateBlackList>
    <!--optional, xs:boolean, Whether to support uploading ID Card blocklist-->
</isSupportUploadCertificateBlackList>
<isSupportGetRegisterInfo>
    <!--optional, xs:boolean, Whether supports getting registered information-->
</isSupportGetRegisterInfo>
<isSupportDownloadCertificateBlackListTemplet>
    <!--optional, xs:boolean, Whether to support downloading template of ID
card blocklist-->
</isSupportDownloadCertificateBlackListTemplet>

```

```

<ScheduleInfo>
    <!-- optional, xs:boolean, shift schedule information-->
    <command opt="personal, everyone">
        <!--required, xs:string, Search condition-->
    </command>
    <employeeNo min="" max="">
        <!--required, xs:integer, Employee No.-->
    </employeeNo>
    <name min="1" max="32">
        <!--required, xs:string, Name-->
    </name>
    <departmentName min="1" max="32">
        <!--required, xs:string, Department name-->
    </departmentName>
    <schedulePlanNo min="" max="">
        <!--required, xs:integer, Shift schedule No.-->
    </schedulePlanNo>
    <schedulePlanType opt="personal, department">
        <!--required, xs:string, Shift schedule type-->
    </schedulePlanType>
    <enabled opt="true, false">
        <!--required, xs:boolean, Enable-->
    </enabled>
    <scheduleType opt="noSchedule, ordinaryClass, workingClass">
        <!--required, xs:string, Shift type-->
    </scheduleType>
    <scheduleNo min="" max="">
        <!--required, xs:integer, Shift No.-->
    </scheduleNo>
    <scheduleStartTime>
        <!--required, xs:time, ISO8601 time, "2016-01-01", Start time-->
    </scheduleStartTime>
    <scheduleEndTime>
        <!--required, xs:time, ISO8601 time, "2016-02-17", End time-->
    </scheduleEndTime>
    <holidayNo min="" max="">
        <!--required, xs:integer, Holiday group No.-->
    </holidayNo>
</ScheduleInfo>
<AttendanceSummaryInfo>
    <!-- optional, xs:boolean, Time and attendance information overview-->
    <command opt="personal, everyone">
        <!--required, xs:string, Search condition-->
    </command>
    <employeeNo min="" max="">
        <!--required, xs:integer, Employee No.-->
    </employeeNo>
    <name min="1" max="32">
        <!--required, xs:string, Name-->
    </name>
    <departmentName min="1" max="32">
        <!--required, xs:string, Department name-->
    </departmentName>

```

```

</departmentName>
<workStandard min="" max="">
    <!--required, xs:integer, Standard working time (minutes)-->
</workStandard>
<workActual min="" max="">
    <!--required, xs:integer, Actual working time (minutes)-->
</workActual>
<lateTimes min="" max="">
    <!--required, xs:integer, Late times-->
</lateTimes>
<lateMinutes min="" max="">
    <!--required, xs:integer, Total late time (minutes)-->
</lateMinutes>
<leaveEarlyTimes min="" max="">
    <!--required, xs:integer, Early Leave Times-->
</leaveEarlyTimes>
<leaveEarlyMinutes min="" max="">
    <!--required, xs:integer, Total early leave time (minutes)-->
</leaveEarlyMinutes>
<overtimeStandard min="" max="">
    <!--required, xs:integer, Standard Overtime (minutes)-->
</overtimeStandard>
<overtimeActual min="" max="">
    <!--required, xs:integer, Actual Overtime (minutes)-->
</overtimeActual>
<attendanceStandard min="" max="">
    <!--required, xs:integer, Standard Attendance (day)-->
</attendanceStandard>
<attendanceActual min="" max="">
    <!--required, xs:integer, Actual Attendance (Day)-->
</attendanceActual>
<absentDays min="" max="">
    <!--required, xs:integer, Absent (Day)-->
</absentDays>
</AttendanceSummaryInfo>
<AttendanceRecordInfo>
    <!--optional, xs:boolean, Time and Attendance Records-->
    <command opt="personal,everyone">
        <!--required, xs:string, Search Condition-->
    </command>
    <employeeNo min="" max="">
        <!--required, xs:integer, Employee No.-->
    </employeeNo>
    <name min="1" max="32">
        <!--required, xs:string, Name-->
    </name>
    <departmentName min="1" max="32">
        <!--required, xs:string, Department Name-->
    </departmentName>
    <attendanceTime>
        <!--required, xs:time, ISO8601 time, "2016-02-17T17:30:08+08:00", Attendance Time-->

```

```

        </attendanceTime>
    </AttendanceRecordInfo>
<AbnormalInfo>
    <!-- optional, xs:boolean, Exception Statistics Information-->
    <command opt="personal,everyone">
        <!--required, xs:string, Search Condition-->
    </command>
    <employeeNo min="" max="">
        <!--required, xs:integer, Employee No.-->
    </employeeNo>
    <name min="1" max="32">
        <!--required, xs:string, Name-->
    </name>
    <departmentName min="1" max="32">
        <!--required, xs:string, Department Name-->
    </departmentName>
    <onDutyTime>
        <!--required, xs:time, ISO8601 time, "2016-02-17T08:30:08+08:00", Start-
Work Time-->
    </onDutyTime>
    <offDutyTime>
        <!--required, xs:time, ISO8601 time, "2016-02-17T17:30:08+08:00", End-
Work Time-->
    </offDutyTime>
    <lateMinutes min="" max="">
        <!--required, xs:integer, Late Duration (minutes)-->
    </lateMinutes>
    <leaveEarlyMinutes min="" max="">
        <!--required, xs:integer, Early Leave Duration (minutes)-->
    </leaveEarlyMinutes>
    <absenceMinutes min="" max="">
        <!--required, xs:integer, Absent Duration (minutes)-->
    </absenceMinutes>
    <totalMinutes min="" max="">
        <!--required, xs:integer, Total Duration (minutes)-->
    </totalMinutes>
</AbnormalInfo>
<CheckFacePicture>
    <!-- optional, xs:boolean, authenticate identity via 1:N face picture
matching-->
    <pictureNum min="" max="">
        <!--required, xs:integer, picture number>
    </pictureNum>
    <checkStatus opt="1,2,3,4,5,6,7,8,9,10,11">
        <!--required, xs:integer, matching result: 1-modeling completed, 2-
modeling failed, 3-the communication with the face picture module failed, 4-no
face in the picture, 5-the face is too close to the top picture border, 6-the
face is too close to the bottom picture border, 7-the face is too close to the
left picture border, 8-the face is too close to the right picture border, 9-the
face picture is clockwise, 10-the face picture is anticlockwise, 11-the
proportion of the pupillary distance is small, 12-face picture matches the
template, 13-face picture mismatches the template>

```

```

</checkStatus>
<checkTemplate opt="0,1">
    <!--optional, xs:integer, 0-picture verification, 1-picture and modeling
data matching verification>
</checkTemplate>
</CheckFacePicture>
<supplementLightNo min="" max="" />
<!--optional, supplement light No.-->
<maxWhiteFaceNum/>
<!--optional, the maximum number of face picture in allowlist>
<maxBlackFaceNum/>
<!--optional, the maximum number of face picture in blocklist>
<isSupportGetFailedFaceInfo>
    <!--optional, xs:boolean, whether supports getting the information of face
modeling failure after upgrading-->
</isSupportGetFailedFaceInfo>
<FailedFaceInfoParam>
    <!--optional, xs:boolean, get the information of face modeling failure
after upgrading-->
    <FailedFaceInfoCond/>
    <FailedFaceInfo>
        <!--required, face modeling failure information-->
        <cardNo min="" max="" />
        <!--required, card number-->
        <errorCode min="" max="" />
        <!--required, face modeling failure error code-->
    </FailedFaceInfo>
<isSupportFaceAndTemplate>
    <!--optional, xs:boolean, whether supports configuring face picture and
modeling data-->
</isSupportFaceAndTemplate>
<FaceAndTemplateParam>
    <!--optional, face picture and modeling data configuration-->
    <cardNo min="" max="" />
    <!--required, card number-->
    <faceLen min="" max="" />
    <!--required, face picture size-->
    <faceTemplateLen min="" max="" />
    <!--required, face picture template data size-->
</FaceAndTemplateParam>
</AcsAbility>

```

## F.178 XML\_CaptureFingerPrint

CaptureFingerPrint message in XML format

```

<CaptureFingerPrint version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <fingerData><!--dep, xs:string, fingerprint data, which is between 1 and 768,
and it should be encoded by Base64--></fingerData>
    <fingerNo><!--req, xs:integer, finger No., which is between 1 and 10--></

```

```
fingerNo>
  <fingerPrintQuality><!--req, xs:integer, fingerprint quality, which is
between 1 and 100--></fingerPrintQuality>
</CaptureFingerPrint>
```

### F.179 XML\_CaptureFingerPrintCond

CaptureFingerPrintCond message in XML format

```
<CaptureFingerPrintCond version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <fingerNo><!--req, xs: integer, finger No., which is between 1 and 10--></
fingerNo>
</CaptureFingerPrintCond>
```

### F.180 XML\_Cap\_AccessControl

AccessControl capability message in XML format

```
<AccessControl version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <isSupportWiegandCfg>
    <!--optional, xs:boolean, whether it supports Wiegand configuration-->
  </isSupportWiegandCfg>
  <isSupportModuleStatus>
    <!--optional, xs:boolean, whether it supports getting the status of secure
door control unit-->
  </isSupportModuleStatus>
  <isSupportSNAPConfig>
    <!--optional, xs:boolean, whether it supports getting capture linkage
parameters-->
  </isSupportSNAPConfig>
  <LocalController><!--opt-->
    <isSupportLocalControllerManage>
      <!--optional, xs:boolean, whether it supports distributed access
controller management-->
    </isSupportLocalControllerManage>
    <isSupportLocalControllerControl>
      <!--optional, xs:boolean, whether it supports distributed access
controller control-->
    </isSupportLocalControllerControl>
  </LocalController>
  <isSupportUSBManage>
    <!--optional, xs:boolean, whether it supports USB management of access
control device-->
  </isSupportUSBManage>
  <isSupportIdentityTerminal>
    <!--optional, xs:boolean, whether it supports face recognition terminal
configuration-->
  </isSupportIdentityTerminal>
```

```
<isSupportDepartmentParam>
    <!--optional, xs:boolean, whether it supports setting department
parameters-->
</isSupportDepartmentParam>
<isSupportSchedulePlan>
    <!--optional, xs:boolean, whether it supports setting shift schedule-->
</isSupportSchedulePlan>
<isSupportAttendanceRule>
    <!--optional, xs:boolean, whether it supports setting time and attendance
rule-->
</isSupportAttendanceRule>
<isSupportOrdinaryClass>
    <!--optional, xs:boolean, whether it supports setting normal shift
parameters-->
</isSupportOrdinaryClass>
<isSupportWorkingClass>
    <!--optional, xs:boolean, whether it supports setting man-hour shift
parameters-->
</isSupportWorkingClass>
<isSupportAttendanceHolidayGroup>
    <!--optional, xs:boolean, whether it supports setting holiday group for
time and attendance-->
</isSupportAttendanceHolidayGroup>
<isSupportAttendanceHolidayPlan>
    <!--optional, xs:boolean, whether it supports setting holiday schedule for
time and attendance-->
</isSupportAttendanceHolidayPlan>
<isSupportLadderControlRelay>
    <!--optional, xs:boolean, whether it supports setting elevator controller
relay-->
</isSupportLadderControlRelay>
<isSupportWiegandRuleCfg>
    <!--optional, xs:boolean, whether it supports setting Wiegand rule-->
</isSupportWiegandRuleCfg>
<isSupportM1CardEncryptCfg>
    <!--optional, xs:boolean, whether it supports M1 card encryption
authentication-->
</isSupportM1CardEncryptCfg>
<isSupportDeployInfo>
    <!--optional, xs:boolean, whether it supports getting arming information-->
</isSupportDeployInfo>
<isSupportSubmarineBack>
    <!--optional, xs:boolean, whether it supports specifying anti-passing back
server-->
</isSupportSubmarineBack>
<isSupportSubmarineBackHostInfo>
    <!--optional, xs:boolean, whether it supports setting access controllers
with anti-passing back enabled-->
</isSupportSubmarineBackHostInfo>
<isSupportStartReaderInfo>
    <!--optional, xs:boolean, whether it supports setting first card reader-->
</isSupportStartReaderInfo>
```

```

<isSupportSubmarineBackReader>
    <!--optional, xs:boolean, whether it supports setting anti-passing back
card reader-->
</isSupportSubmarineBackReader>
<isSupportServerDevice>
    <!--optional, xs:boolean, whether it supports setting anti-passing back
server information-->
</isSupportServerDevice>
<isSupportReaderAcrossHost>
    <!--optional, xs:boolean, whether it supports enabling cross-controller
anti-passing back function of card reader-->
</isSupportReaderAcrossHost>
<isSupportClearCardRecord>
    <!--optional, xs:boolean, whether it supports clearing card swiping records
in anti-passing back server-->
</isSupportClearCardRecord>
<isSupportSubmarineBackMode>
    <!--optional, xs:boolean, whether it supports setting anti-passing back
mode-->
</isSupportSubmarineBackMode>
<isSupportClearSubmarineBack>
    <!--optional, xs:boolean, whether it supports clearing cross-controller
anti-passing back information-->
</isSupportClearSubmarineBack>
<isSupportFaceCompareCond><!--optional, xs:boolean, whether it supports
configuring restriction condition parameters of face picture comparison--></
isSupportFaceCompareCond>
<isSupportRemoteControlDoor>
    <!--optional, xs:boolean, whether it supports remote door, elevator, and
lock control: "true"-yes, this node is not returned-no-->
</isSupportRemoteControlDoor>
<isSupportUserInfo><!--optional, xs:boolean, whether it supports person
management based on person--></isSupportUserInfo>
<EmployeeNoInfo><!--dep, employee No. (person ID) information, this node is
valid only when the isSupportUserInfo is "true"-->
    <employeeNo min="" max=""><!--optional, employee No. (person ID)--></
employeeNo>
    <characterType opt="any,number">
        <!--optional, employee No. (person) ID type: "any"-any characters
(default), "number"-digits (from 0 to 9), only one value can be returned-->
    </characterType>
    <isSupportCompress>
        <!--optional, xs:boolean, whether it supports compressing employee No.
(person ID) for storage: "true"-yes, this node is not returned-no-->
    </isSupportCompress>
</EmployeeNoInfo>
<isSupportCardInfo><!--optional, xs:boolean, whether it supports card
management based on person: "true"-yes, this node is not returned-no--></
isSupportCardInfo>
<isSupportFDLib><!--optional, xs:boolean, whether it supports face picture
library management--></isSupportFDLib>
<isSupportUserInfoDetailDelete><!--optional, xs:boolean, whether it supports

```

```
deleting person information and permission: "true"-yes, this node is not
returned-no--></isSupportUserInfoDetailDelete>
<isSupportAuthCodeInfo>
    <!--optional, xs:boolean, whether it supports authentication password
management: "true"-yes, this node is not returned-no-->
</isSupportAuthCodeInfo>
<isSupportFingerPrintCfg>
    <!--optional, xs:boolean, whether it supports configuring fingerprint
parameters: "true"-yes, this node is not returned-no-->
</isSupportFingerPrintCfg>
<isSupportFingerPrintDelete>
    <!--optional, xs:boolean, whether it supports deleting fingerprint: "true"-yes,
this node is not returned-no-->
</isSupportFingerPrintDelete>
<isSupportCaptureFingerPrint>
    <!--optional, xs:boolean, whether it supports collecting fingerprint
information: "true"-yes, this node is not returned-no-->
</isSupportCaptureFingerPrint>
<isSupportDoorStatusWeekPlanCfg>
    <!--optional, xs:boolean, whether it supports configuring door control week
schedule: "true"-yes, this node is not returned-no-->
</isSupportDoorStatusWeekPlanCfg>
<isSupportVerifyWeekPlanCfg>
    <!--optional, xs:boolean, whether it supports configuring week schedule of
the card reader authentication mode: "true"-yes, this node is not returned-no-->
</isSupportVerifyWeekPlanCfg>
<isSupportCardRightWeekPlanCfg>
    <!--optional, xs:boolean, whether it supports configuring week schedule of
the access permission control: "true"-yes, this node is not returned-no-->
</isSupportCardRightWeekPlanCfg>
<isSupportDoorStatusHolidayPlanCfg>
    <!--optional, xs:boolean, whether it supports configuring door control
holiday schedule: "true"-yes, this node is not returned-no-->
</isSupportDoorStatusHolidayPlanCfg>
<isSupportVerifyHolidayPlanCfg>
    <!--optional, xs:boolean, whether it supports configuring holiday schedule
of the card reader authentication mode: "true"-yes, this node is not returned-
no-->
</isSupportVerifyHolidayPlanCfg>
<isSupportCardRightHolidayPlanCfg>
    <!--optional, xs:boolean, whether it supports configuring holiday schedule
of the access permission control: "true"-yes, this node is not returned-no-->
</isSupportCardRightHolidayPlanCfg>
<isSupportDoorStatusHolidayGroupCfg>
    <!--optional, xs:boolean, whether it supports configuring holiday group of
the door control schedule: "true"-yes, this node is not returned-no-->
</isSupportDoorStatusHolidayGroupCfg>
<isSupportVerifyHolidayGroupCfg>
    <!--optional, xs:boolean, whether it supports configuring holiday group of
the control schedule of the card reader authentication mode: "true"-yes, this
node is not returned-no-->
</isSupportVerifyHolidayGroupCfg>
```

```
<isSupportUserRightHolidayGroupCfg>
    <!--optional, xs:boolean, whether it supports configuring holiday group of
the access permission control schedule: "true"-yes, this node is not returned-
no-->
</isSupportUserRightHolidayGroupCfg>
<isSupportDoorStatusPlanTemplate>
    <!--optional, xs:boolean, whether it supports configuring door control
schedule template: "true"-yes, this node is not returned-no-->
</isSupportDoorStatusPlanTemplate>
<isSupportVerifyPlanTemplate>
    <!--optional, xs:boolean, whether it supports configuring schedule template
of the card reader authentication mode: "true"-yes, this node is not returned-
no-->
</isSupportVerifyPlanTemplate>
<isSupportUserRightPlanTemplate>
    <!--optional, xs:boolean, whether it supports configuring schedule template
of the access permission control: "true"-yes, this node is not returned-no-->
</isSupportUserRightPlanTemplate>
<isSupportDoorStatusPlan>
    <!--optional, xs:boolean, whether it supports configuring door control
schedule: "true"-yes, this node is not returned-no-->
</isSupportDoorStatusPlan>
<isSupportCardReaderPlan>
    <!--optional, xs:boolean, whether it supports configuring control schedule
of the card reader authentication mode: "true"-yes, this node is not returned-
no-->
</isSupportCardReaderPlan>
<isSupportClearPlansCfg>
    <!--optional, xs:boolean, whether it supports clearing the access control
schedule parameters: "true"-yes, this node is not returned-no-->
</isSupportClearPlansCfg>
<isSupportRemoteControlBuzzer>
    <!--optional, xs:boolean, whether it supports remotely controlling the
buzzer of the card reader: "true"-yes, this node is not returned-no-->
</isSupportRemoteControlBuzzer>
<isSupportEventCardNoList>
    <!--optional, xs:boolean, whether it supports getting the list of event and
card linkage ID: "true"-yes, this node is not returned-no-->
</isSupportEventCardNoList>
<isSupportEventCardLinkageCfg>
    <!--optional, xs:boolean, whether it supports configuring event and card
linkage parameters: "true"-yes, this node is not returned-no-->
</isSupportEventCardLinkageCfg>
<isSupportClearEventCardLinkageCfg>
    <!--optional, xs:boolean, whether it supports clearing event and card
linkage parameters: "true"-yes, this node is not returned-no-->
</isSupportClearEventCardLinkageCfg>
<isSupportAcsEvent>
    <!--optional, xs:boolean, whether it supports searching for access control
events: "true"-yes, this node is not returned-no-->
</isSupportAcsEvent>
<isSupportAcsEventTotalNum>
```

```
<!--optional, xs:boolean, whether it supports getting total number of
access control events by specific conditions: "true"-yes, this node is not
returned-no-->
</isSupportAcsEventTotalNum>
<isSupportDeployInfo>
    <!--optional, xs:boolean, whether it supports getting the arming
information: "true"-yes, this node is not returned-no-->
    </isSupportDeployInfo>
    <isSupportEventOptimizationCfg>
        <!--optional, xs:boolean, whether it supports configuring event
optimization: "true"-yes, this node is not returned-no-->
        </isSupportEventOptimizationCfg>
    <isSupportAcsWorkStatus>
        <!--optional, xs:boolean, whether it supports getting working status of the
access control device: "true"-yes, this node is not returned-no-->
        </isSupportAcsWorkStatus>
    <isSupportDoorCfg>
        <!--optional, xs:boolean, whether it supports configuring door parameters:
"true"-yes, this node is not returned-no-->
        </isSupportDoorCfg>
    <isSupportCardReaderCfg>
        <!--optional, xs:boolean, whether it supports configuring card reader
parameters: "true"-yes, this node is not returned-no-->
        </isSupportCardReaderCfg>
    <isSupportAcsCfg>
        <!--optional, xs:boolean, whether it supports configuring parameters of
access control device: "true"-yes, this node is not returned-no-->
        </isSupportAcsCfg>
    <isSupportRemoteCheck>
        <!--optional, xs:boolean, whether it supports verifying access control
events remotely: true-yes, this field is not returned-no-->
        </isSupportRemoteCheck>
    <isSupportMaskDetection>
        <!--optional, xs:boolean, whether it supports mask detection: true-yes,
this field is not returned-no-->
        </isSupportMaskDetection>
    <isSupportGroupCfg>
        <!--optional, xs:boolean, whether it supports configuring group parameters:
"true"-yes, this node is not returned-no-->
        </isSupportGroupCfg>
    <isSupportClearGroupCfg>
        <!--optional, xs:boolean, whether it supports clearing group parameters:
"true"-yes, this node is not returned-no-->
        </isSupportClearGroupCfg>
    <isSupportMultiCardCfg>
        <!--optional, xs:boolean, whether it supports configuring multiple
authentication mode: "true"-yes, this node is not returned-no-->
        </isSupportMultiCardCfg>
    <isSupportMultiDoorInterLockCfg>
        <!--optional, xs:boolean, whether it supports configuring multi-door
interlocking parameters: "true"-yes, this node is not returned-no-->
        </isSupportMultiDoorInterLockCfg>
```

```

<isSupportAntiSneakCfg>
    <!--optional, xs:boolean, whether it supports configuring anti-passing back
parameters in the device: "true"-yes, this node is not returned-no-->
</isSupportAntiSneakCfg>
<isSupportCardReaderAntiSneakCfg>
    <!--optional, xs:boolean, whether it supports configuring anti-passing back
parameters for the card reader in the device: "true"-yes, this node is not
returned-no-->
</isSupportCardReaderAntiSneakCfg>
<isSupportClearAntiSneakCfg>
    <!--optional, xs:boolean, whether it supports clearing anti-passing back
parameters: "true"-yes, this node is not returned-no-->
</isSupportClearAntiSneakCfg>
<isSupportClearAntiSneak>
    <!--optional, xs:boolean, whether it supports clearing anti-passing back
records in the device: "true"-yes, this node is not returned-no-->
</isSupportClearAntiSneak>
<isSupportSmsRelativeParam>
    <!--optional, xs:boolean, whether it supports configuring message function:
"true"-yes, this node is not returned-no-->
</isSupportSmsRelativeParam>
<isSupportPhoneDoorRightCfg>
    <!--optional, xs:boolean, whether it supports configuring the door
permission linked to the mobile phone number: "true"-yes, this node is not
returned-no-->
</isSupportPhoneDoorRightCfg>
<isSupportOSDPStatus>
    <!--optional, xs:boolean, whether it supports searching for OSDP card
reader status: "true"-yes, this node is not returned-no-->
</isSupportOSDPStatus>
<isSupportOSDPMModify>
    <!--optional, xs:boolean, whether it supports editing OSDP card reader ID:
"true"-yes, this node is not returned-no-->
</isSupportOSDPMModify>
<isSupportLogModeCfg>
    <!--optional, xs:boolean, whether it supports configuring log mode: "true"-yes,
this node is not returned-no-->
</isSupportLogModeCfg>
<FactoryReset>
    <isSupportFactoryReset><!--optional, xs: boolean, whether it supports
restoring to default settings by condition--></isSupportFactoryReset>
    <mode opt="full,basic,part"><!--optional, xs: string, conditions for
restoring to default settings--></mode>
</FactoryReset>
<isSupportNFCCfg><!--optional, xs:boolean, whether it supports enabling or
disabling NFC function: "true"-yes, this node is not returned-no--></
isSupportNFCCfg>
<isSupportRFCardCfg><!--optional, xs:boolean, whether it supports enabling or
disabling RF card recognition: "true"-yes, this node is not returned-no--></
isSupportRFCardCfg>
<isSupportCaptureFace>
    <!--optional, xs:boolean, whether it supports collecting face pictures:

```

```

"true"-yes, this node is not returned-no-->
</isSupportCaptureFace>
<isSupportCaptureInfraredFace>
    <!--optional, xs:boolean, whether it supports collecting infrared face
pictures: "true"-yes, this node is not returned-no-->
</isSupportCaptureInfraredFace>
<isSupportFaceRecognizeMode>
    <!--optional, xs:boolean, whether it supports configuring facial
recognition mode: "true"-yes, this node is not returned-no-->
</isSupportFaceRecognizeMode>
<isSupportRemoteControlPWChcek>
    <!--optional, xs:boolean, whether it supports verifying the password for
remote door control: "true"-yes, this node is not returned-no-->
</isSupportRemoteControlPWChcek>
<isSupportRemoteControlPWCfg>
    <!--optional, xs:boolean, whether it supports configuring the password for
remote door control: "true"-yes, this node is not returned-no-->
</isSupportRemoteControlPWCfg>
<isSupportAttendanceStatusModeCfg>
    <!--optional, xs:boolean, whether it supports configuring attendance mode:
"true"-yes, this node is not returned-no-->
</isSupportAttendanceStatusModeCfg>
<isSupportAttendanceStatusRuleCfg>
    <!--optional, xs:boolean, whether it supports configuring attendance status
and rule: "true"-yes, this node is not returned-no-->
</isSupportAttendanceStatusRuleCfg>
<isSupportCaptureCardInfo>
    <!--optional, xs:boolean, whether it supports collecting card information:
"true"-yes, this node is not returned-no-->
</isSupportCaptureCardInfo>
<isSupportCaptureIDInfo>
    <!--optional, xs:boolean, whether it supports collecting ID card
information: "true"-yes, this node is not returned-no-->
</isSupportCaptureIDInfo>
<isSupportCaptureRule>
    <!--optional, xs:boolean, whether it supports configuring online collection
rules: "true"-yes, this node is not returned-no-->
</isSupportCaptureRule>
<isSupportCapturePresetParam>
    <!--optional, xs:boolean, whether it supports configuring preset parameters
of online collection: "true"-yes, this node is not returned-no-->
</isSupportCapturePresetParam>
<isSupportOfflineCapture>
    <!--optional, xs:boolean, whether it supports offline collection: "true"-yes,
this node is not returned-no-->
</isSupportOfflineCapture>
<isSupportCardOperations>
    <!--optional, xs:boolean, whether it supports card operation: "true"-yes,
this node is not returned-no-->
</isSupportCardOperations>
<isSupportRightControllerAudio>
    <!--optional, xs:boolean, whether it supports configuring audio file

```

```

parameters of the main controller-->
</isSupportRightControllerAudio>
<isSupportChannelControllerCfg>
    <!--optional, xs:boolean, whether it supports configuring lane controller-->
</isSupportChannelControllerCfg>
<isSupportGateDialAndInfo>
    <!--optional, xs:boolean, whether it supports getting local DIP and
information of the turnstile-->
</isSupportGateDialAndInfo>
<isSupportGateStatus>
    <!--optional, xs:boolean, whether it supports getting turnstile status-->
</isSupportGateStatus>
<isSupportGateIRStatus>
    <!--optional, xs:boolean, whether it supports getting the status of the
active infrared intrusion detector of the turnstile-->
</isSupportGateIRStatus>
<isSupportGateRelatedPartsStatus>
    <!--optional, xs:boolean, whether it supports getting related components'
status of the turnstile-->
</isSupportGateRelatedPartsStatus>
<isSupportChannelControllerAlarmLinkage>
    <!--optional, xs:boolean, whether it supports configuring alarm linkage of
the lane controller-->
</isSupportChannelControllerAlarmLinkage>
<isSupportChannelControllerAlarmOut>
    <!--optional, xs:boolean, whether it supports configuring alarm output of
the lane controller-->
</isSupportChannelControllerAlarmOut>
<isSupportChannelControllerAlarmOutControl>
    <!--optional, xs:boolean, whether it supports controlling alarm output of
the lane controller-->
</isSupportChannelControllerAlarmOutControl>
<isSupportChannelControllerTypeCfg>
    <!--optional, xs:boolean, whether it supports configuring device type of
the lane controller-->
</isSupportChannelControllerTypeCfg>
<isSupportRemoteCtrlrModeCfg>
    <!--optional, xs:boolean, whether it supports configuring parameters of the
keyfob control mode-->
</isSupportRemoteCtrlrModeCfg>
<isSupportTTSText><!--optional, xs:boolean, whether it supports configuring
the text of the audio prompt: true-yes. If this function is not supported, this
node will be not returned--></isSupportTTSText>
<isSupportIDBlackListCfg><!--optional, xs:boolean, whether it supports
applying ID card blocklist: true-yes. If this function is not supported, this
node will be not returned--></isSupportIDBlackListCfg>
<isSupportUserDataImport><!--optional, xs:boolean, whether it supports
importing person permission data: true-yes. If this function is not supported,
this node will be not returned--></isSupportUserDataImport>
<isSupportUserDataExport><!--optional, xs:boolean, whether it supports
exporting person permission data: true-yes. If this function is not supported,
this node will be not returned--></isSupportUserDataExport>
```

```

<isSupportMaintenanceDataExport><!--optional, xs:boolean, whether it supports
exporting maintenance data: true-yes. If this function is not supported, this
node will be not returned--></isSupportMaintenanceDataExport>
<isSupportLockTypeCfg><!--optional, xs:boolean, whether it supports
configuring door lock status when the device is powered off: true-yes. If this
function is not supported, this node will be not returned--></
isSupportLockTypeCfg>
<isSupportSafetyHelmetDetection><!--optional, xs:boolean, whether it supports
configuring hard hat detection: true-yes, this node is not returned-no--></
isSupportSafetyHelmetDetection>
<isSupportKeyCfgAttendance><!--optional, xs:boolean, whether it supports
configuring parameters of attendance check by pressing the key: true-yes, this
node is not returned-no--></isSupportKeyCfgAttendance>
<isSupportIDBlackListTemplate><!--optional, xs:boolean, whether it supports
downloading the ID card blocklist template: true-yes, this node is not returned-
no--></isSupportIDBlackListTemplate>
<isSupportAttendanceWeekPlan><!--optional, xs:boolean, whether it supports
configuring parameters of the week attendance schedule: true-yes, this node is
not returned-no--></isSupportAttendanceWeekPlan>
<isSupportClearAttendancePlan><!--optional, xs:boolean, whether it supports
clearing the week attendance schedule: true-yes, this node is not returned-no-->
</isSupportClearAttendancePlan>
<isSupportAttendanceMode><!--optional, xs:boolean, whether it supports
configuring the attendance mode: true-yes, this node is not returned-no--></
isSupportAttendanceMode>
<isSupportAttendancePlanTemplate><!--whether it supports configuring the
attendance schedule template: true-yes, this node is not returned-no--></
isSupportAttendancePlanTemplate>
<isSupportAttendancePlanTemplateList><!--optional, xs:boolean, whether it
supports getting the list of attendance schedule templates: true-yes, this node
is not returned-no--></isSupportAttendancePlanTemplateList>
<isSupportCardVerificationRule><!--optional, xs:boolean, whether it supports
configuring card No. authentication mode: true-yes, this node is not returned-
no--></isSupportCardVerificationRule>
<isSupportTemperatureMeasureCfg><!--optional, xs:boolean, whether it
supports configuring temperature measurement parameters: true (support), this
node is not returned (not support)--></isSupportTemperatureMeasureCfg>
<isSupportTemperatureMeasureAreaCfg><!--optional, xs:boolean, whether it
supports configuring parameters of the temperature measurement area: true
(support), this node is not returned (not support)--></
isSupportTemperatureMeasureAreaCfg>
<isSupportTemperatureMeasureAreaCalibrationCfg><!--optional, xs:boolean,
whether it supports configuring calibration parameters of the temperature
measurement area: true (support), this node is not returned (not support)--></
isSupportTemperatureMeasureAreaCalibrationCfg>
<isSupportBlackObjectCfg><!--optional, xs:boolean, whether it supports
configuring black body parameters: true (support), this node is not returned
(not support)--></isSupportBlackObjectCfg>
<isSupportHealthCodeCfg><!--optional, xs:boolean, whether it supports
configuring health code parameters: true (support), this node is not returned
(not support)--></isSupportHealthCodeCfg>
<isSupportShowHealthCodeCfg><!--optional, xs:boolean, whether it supports

```

```
configuring display parameters of the health code: true (support), this node is
not returned (not support)--></isSupportShowHealthCodeCfg>
<isSupportAddCustomAudio><!--optional, boolean, whether it supports importing
custom audio, related URI: /ISAPI/AccessControl/customAudio/addCustomAudio?
format=json--></isSupportAddCustomAudio>
<isSupportDeleteCustomAudio><!--optional, boolean, whether it supports
deleting custom audio, related URI: /ISAPI/AccessControl/customAudio/
deleteCustomAudio?format=json--></isSupportDeleteCustomAudio>
<isSupportSearchCustomAudio><!--optional, boolean, whether it supports
searching for custom audio, related URI: /ISAPI/AccessControl/customAudio/
searchCustomAudioStatus?format=json--></isSupportSearchCustomAudio>
<isSupportBluetoothEncryptionInfo><!--optional, xs:boolean, whether it
supports configuring bluetooth encryption information: true (support). If this
function is not supported, this node will not be returned--></
isSupportBluetoothEncryptionInfo>
<isSupportBluetoothEncryptionVersion><!--optional, xs:boolean, whether it
supports configuring bluetooth encryption version: true (support). If this
function is not supported, this node will not be returned--></
isSupportBluetoothEncryptionVersion>
<isSupportBluetooth><!--optional, xs:boolean, whether it supports bluetooth
configuration--></isSupportBluetooth>
</AccessControl>
```

## F.181 XML\_Cap\_CaptureFingerPrint

CaptureFingerPrint capability message in XML format

```
<CaptureFingerPrint version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <CaptureFingerPrintCond><!--req, xs: integer, finger No.-->
    <fingerNo min="1" max="10"></fingerNo>
  </CaptureFingerPrintCond>
  <fingerData min="1" max="768"><!--dep, xs:string, fingerprint data--></
fingerData>
  <fingerNo min="1" max="10"><!--req, xs:integer, finger No.--></fingerNo>
  <fingerPrintQuality min="1" max="100"><!--req, xs:integer, fingerprint
quality--></fingerPrintQuality>
</CaptureFingerPrint>
```

## F.182 XML\_Cap\_ChannelControllerCfg

XML message about the configuration capability of lane controller

```
<ChannelControllerCfg version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
  <gatePassingMode opt="ByChannelController,ByRightController"><!--required,
xs:string, turnstile passing mode: "ByChannelController"-based on the lane
controller's local DIP settings, "ByRightController"-based on the main
controller's settings--></gatePassingMode>
  <freePassAuthEnabled opt="enable,disable"><!--required, xs:string, whether
```

```

the authentication is required for free passing: "enable"-yes, "disable"-no--></
freePassAuthEnabled>
    <openAndCloseSpeed min="1" max="10"><!--required, xs:integer, barrier's
opening and closing speed, it is between 1 and 10, which represents the speed
from 10% to 100%--></openAndCloseSpeed>
        <alarmSoundTime min="0" max="599"><!--required, xs:integer, alarm prompt
sound duration, unit: second. The value is between 0 and 599, and 0 refers to
continuously playing alarm prompt sound--></alarmSoundTime>
            <tempUnit opt="Centigrade,Fahrenheit"><!--required, xs:string, temperature
unit to be displayed: "Centigrade"-Celsius (°C), "Fahrenheit"-Fahrenheit (°F)--></tempUnit>
                <alarmAreaNoAuth opt="true,false"><!--optional, xs:boolean, whether opening
door is prohibited in the alarm area--></alarmAreaNoAuth>
                    <gateWingMaterial opt="Acrylic,StellPipe,SinglePUGate,DoublePUGate"><!--
optional, xs:string, barrier material: "Acrylic"-acrylic, "StellPipe"-steel
pipe, "SinglePUGate"-single PU gate, "DoublePUGate"-two PU gates--></
gateWingMaterial>
                        <channelLength min="550" max="1400"><!--optional, xs:integer, barrier length,
unit: mm--></channelLength>
                            <motorDirection opt="Clockwise,AntiClockwise"><!--optional, xs:string, motor
rotation direction: "Clockwise", "AntiClockwise"--></motorDirection>
                                <lampBoardLight min="" max=""><!--optional, xs:integer, light board
brightness, it is between 0 and 100--></lampBoardLight>
                                    <openSpeed min="" max=""><!--optional, xs:string, barrier's opening speed, it
is between 1 and 10 which represents the speed from 10% to 100%, and the
default speed is 50%. If openAndCloseSpeed and openSpeed are both configured,
the barrier's opening speed is determined by openSpeed--></openSpeed>
                                        <closeSpeed min="1" max="10"><!--optional, xs:integer, barrier's closing
speed, it is between 1 and 10 which represents the speed from 10% to 100%, and
the default speed is 40%. If openAndCloseSpeed and closeSpeed are both
configured, the barrier's closing speed is determined by closeSpeed--></
closeSpeed>
                                            <runMode><!--optional, xs:string, running mode: "doubleGateWing"-two barriers
mode (default), "singleGateWing"-single barrier mode--></runMode>
</ChannelControllerCfg>

```

### F.183 XML\_Cap\_ClearCardRecord

ClearCardRecord capability message in XML format

```

<ClearCardRecord version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <clearAllCard opt="true,false">
        <!--req, xs: boolean, whether to clear all card swiping records in the
cross-controller anti-passing back server-->
    </clearAllCard>
    <CardList size="32">
        <cardNo min="1" max="32"><!--opt, xs: string, card No.--></cardNo>
    </CardList>
    <EmployeeNoList size="32">
        <employeeNo min="" max=""><!--opt, xs:string, employee No. (person ID)--></

```

```
employeeNo>
  </EmployeeNoList>
</ClearCardRecord>
```

### F.184 XML\_Cap\_ClearSubmarineBack

ClearSubmarineBack capability message in XML format

```
<ClearSubmarineBack version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <clearHostInfo opt="true,false"><!--opt, xs: boolean, whether to clear access controller information--></clearHostInfo>
  <clearReaderInfo opt="true,false"><!--opt, xs: boolean, whether to clear card reader information--></clearReaderInfo>
  <clearSubmarineBack opt="true,false"><!--opt, xs: boolean, whether to clear anti-passing back server parameters--></clearSubmarineBack>
  <clearSubmarineBackHostInfo opt="true,false">
    <!--opt, xs: boolean, whether to clear cross-controller anti-passing back parameters of access controllers-->
  </clearSubmarineBackHostInfo>
  <clearStartReaderInfo opt="true,false"><!--opt, xs: boolean, whether to clear first card reader parameters--></clearStartReaderInfo>
  <clearSubmarineBackReader opt="true,false">
    <!--opt, xs: boolean, whether to clear cross-controller anti-passing back parameters of card readers-->
  </clearSubmarineBackReader>
  <clearSubmarineBackMode opt="true,false">
    <!--opt, xs: boolean, whether to clear the cross-controller anti-passing back mode parameters-->
  </clearSubmarineBackMode>
  <clearServerDevice opt="true,false"><!--opt, xs: boolean, whether to clear the parameters of cross-controller anti-passing back server--></clearServerDevice>
  <clearReaderAcrossHost opt="true,false">
    <!--opt, xs: boolean, whether to clear the cross-controller anti-passing back status of card readers-->
  </clearReaderAcrossHost>
</ClearSubmarineBack>
```

### F.185 XML\_Cap\_GetAcsEvent

GetAcsEvent capability message in XML format

```
<GetAcsEvent version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <EventCond><!--req, event search conditions-->
    <majorType></majorType><!--req, event major type, see details in Access Control Event Types-->
      <minorType></minorType><!--req, event minor type, see details in Access Control Event Types-->
        <startTime></startTime><!--req, start time-->
```

```
<endTime></endTime><!--req, end time-->
<localOrUTC></localOrUTC><!--opt, time type: "Local"-device local time
(default), "UTC"-UTC time. If this node is not returned, the startTime and
endTime will be used as the local time by default-->
<cardNo min="" max=""></cardNo><!--req, card No.-->
<name min="" max=""></name><!--req, cardholder name-->
<picEnable opt="true,false"></picEnable><!--req, whether contains picture-->
<beginSerialNo min="" max=""></beginSerialNo><!--req, start serial No.-->
<endSerialNo min="" max=""></endSerialNo><!--req, end serial No.-->
<employeeNo min="" max=""></employeeNo><!--opt, employee No. (person ID)-->
</EventCond>
<EventLog>
    <majorType>0x1</majorType><!--req, alarm event-->
    <MinorTypeList>
        <minorType>0x400</minorType><!--req, Zone short circuit attempts alarm-->
        <minorType>0x401</minorType><!--req, Zone open circuit attempts alarm-->
        <!--See more minor types of alarm event in Access Control Event Types-->
        <MinorTypeList>
    </EventLog>
    <EventLog>
        <majorType>0x2</majorType><!--req, exception alarm-->
        <MinorTypeList>
            <minorType>0x27</minorType><!--req, Network disconnected-->
            <minorType>0x3a</minorType><!--req, Connection exception-->
            <!--See more minor types of exception event in Access Control Event
Types-->
            <MinorTypeList>
        </EventLog>
        <EventLog>
            <majorType>0x3</majorType><!--req, operation event-->
            <MinorTypeList>
                <minorType>0x400</minorType><!--req, Remotely opened door-->
                <minorType>0x401</minorType><!--req, remotely closed door-->
                <!--See more minor types of operation event in Access Control Event
Types-->
                <MinorTypeList>
            </EventLog>
            <EventLog>
                <majorType>0x5</majorType><!--req, other event-->
                <MinorTypeList>
                    <minorType>0x01</minorType><!--req, Authenticated by valid card-->
                    <minorType>0x02</minorType><!--req, Authenticated by card and password-->
                    <!--See more minor types of other event in Access Control Event Types-->
                    <MinorTypeList>
                </EventLog>
            </EventLog>
        </GetAcsEvent>
```

## See Also

### [Access Control Event Types](#)

## F.186 XML\_Cap\_DeployInfo

DeployInfo capability message in XML format

```
<DeployInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <DeployList size="5">
    <Content>
      <deployNo min="" max=""><!--req, xs: integer, arming No.--></deployNo>
      <deployType opt="0,1,2"><!--req, xs: integer, arming type: 0-client
arming to receive real-time or offline events via platform or system (based on
private protocol), 1-real-time arming to receive real-time events (based on
private protocol), 2-arm based on ISAPI protocol--></deployType>
      <ipAddr min="" max=""><!--req, xs: string, IP address--></ipAddr>
    </Content>
  </DeployList>
</DeployInfo>
```

## F.187 XML\_Cap\_FaceCompareCond

XML message about condition configuration capability of face picture comparison

```
<FaceCompareCond version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <faceWidthLowerLimit min="" max=""><!--optional, xs:integer, face width
threshold with highest priority, value range: [0, 100], when the detected face
width is larger than this threshold, the following conditions will be ignored
and the face comparison will be executed--></faceWidthLowerLimit>
  <pitch min="" max=""><!--optional, xs:integer, face raising or bowing angle,
value range: [0, 90], unit: degree, the smaller the better--></pitch>
  <yaw min="" max=""><!--optional, xs:integer, face siding left or right angle,
value range: [0, 90], unit: degree, the smaller the better--></yaw>
  <width min="" max=""><!--optional, xs:integer, face width, value range: [0,
100]--></width>
  <height min="" max=""><!--optional, xs:integer, face height, value range: [0,
100]--></height>
  <leftBorder min="" max=""><!--optional, xs:integer, left border of face,
value range: [0, 100]--></leftBorder>
  <rightBorder min="" max=""><!--optional, xs:integer, right border of face,
value range: [0, 100]--></rightBorder>
  <upBorder min="" max=""><!--optional, xs:integer, top border of face, value
range: [0, 100]--></upBorder>
  <bottomBorder min="" max=""><!--optional, xs:integer, bottom border of face,
value range: [0, 100]--></bottomBorder>
  <interorbitalDistance min="" max=""><!--optional, xs:integer, pupil distance,
value range: [0, 100]--></interorbitalDistance>
  <faceScore min="" max=""><!--optional, xs:integer, face score, value range:
[0, 100], the valid face score must be larger than this score--></faceScore>
  <maxDistance opt="0.5,1,1.5,2,auto"><!--optional, xs:string, maximum
recognition distance: "0.5,1,1.5,2,auto", unit: m. This node has higher
priority over <interorbitalDistance>--></maxDistance>
```

```

<similarity min="0.0" max="1.0"><!--optional, xs:float, face comparison
similarity--></similarity>
</FaceCompareCond>
```

### F.188 XML\_Cap\_IdentityTerminal

IdentityTerminal capability message in XML format

```

<IdentityTerminal version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <terminalMode opt="authMode,registerMode">
        <!--req, xs: string, terminal mode: "authMode"-authentication mode,
"registerMode"-registration mode-->
    </terminalMode>
    <idCardReader opt="iDR210,DS-K1F110-I,DS-K1F1110-B, DS-K1F1110-AB, none">
        <!--req, xs: string, ID card reader model-->
    </idCardReader>
    <camera opt="C270,DS-2CS5432B-S"><!--req, xs: string, camera--></camera>
    <fingerPrintModule opt="ALIWARD,HikModule"><!--req, xs: string, fingerprint
module--></fingerPrintModule>
    <videoStorageTime min="0" max="10"><!--req, xs: integer, time for saving
video (unit: second)--></videoStorageTime>
    <faceContrastThreshold min="0" max="100"><!--req, xs: integer, face picture
comparison threshold--></faceContrastThreshold>
    <twoDimensionCode opt="enable,disable"><!--req, xs: string, whether to enable
QR code recognition--></twoDimensionCode>
    <blackListCheck opt="enable,disable"><!--req, xs: string, whether to enable
blocklist verification--></blackListCheck>
    <idCardCheckCenter opt="local,server">
        <!--req, xs: string, ID card comparison mode: local-compare with ID card of
local storage, server-compare with ID card of remote server storage-->
    </idCardCheckCenter>
    <faceAlgorithm opt="HIK-Z,HIK-H">
        <!--req, xs: string, face picture algorithm: HIK-Z-Private algorithm, HIK-
third-party algorithm-->
    </faceAlgorithm>
    <comNo min="1" max="9"><!--req, xs: integer, COM No.--></comNo>
    <memoryLearning opt="enable,disable"><!--req, xs: string, whether to enable
learning and memory function--></memoryLearning>
    <saveCertifiedImage opt="enable,disable"><!--req, xs: string, whether to
enable saving authenticated picture--></saveCertifiedImage>
    <MCUVersion min="" max=""><!--opt, xs: string, MCU version information--></
MCUVersion>
    <usbOutput opt="enable,disable"><!--req, xs: string, whether to enable USB
output of ID card reader--></usbOutput>
    <serialOutput opt="enable,disable"><!--req, xs: string, whether to enable
serial port output of ID card reader--></serialOutput>
    <readInfoOfCard opt="serialNo,file"><!--opt, xs: string, set content to be
read from CPU card--></readInfoOfCard>
    <workMode opt="passMode,accessControlMode"><!--opt, xs: string,
authentication mode--></workMode>
```

```

<ecoMode>
    <eco opt="enable,disable"><!--opt, xs: string, whether to enable ECO mode--></eco>
        <faceMatchThreshold1 min="" max=""><!--req, xs: integer, 1V1 face picture comparison threshold of ECO mode, which is between 0 and 100--></faceMatchThreshold1>
        <faceMatchThresholdN min="" max=""><!--req, xs: integer, 1:N face picture comparison threshold of ECO mode, which is between 0 and 100--></faceMatchThresholdN>
        <changeThreshold min="" max=""><!--opt, xs: string, switching threshold of ECO mode, which is between 0 and 8--></changeThreshold>
        <maskFaceMatchThresholdN min="0" max="100"><!--req, xs:integer, 1:N face picture (face with mask and normal background picture) comparison threshold of ECO mode, value range: [0,100]--></maskFaceMatchThresholdN>
        <maskFaceMatchThreshold1 min="0" max="100"><!--req, xs:integer, 1:1 face picture (face with mask and normal background picture) comparison threshold of ECO mode, value range: [0,100]--></maskFaceMatchThreshold1>
    </ecoMode>
    <readCardRule opt="wiegand26,wiegand34"><!--opt, xs: string, card No. setting rule: "wiegand26", "wiegand34"--></readCardRule>
    <enableScreenOff opt="true,false"><!--optional, xs:boolean, whether the device enters the sleep mode when there is no operation after the configured sleep time--></enableScreenOff>
    <screenOffTimeout min="" max=""><!--dependent, xs:integer, sleep time, unit: second--></screenOffTimeout>
    <enableScreensaver opt="true,false"><!--optional, xs:boolean, whether to enable the screen saver function--></enableScreensaver>
    <showMode opt="concise,normal"><!--optional, xs:string, display mode: "concise" (simple mode, only the authentication result will be displayed), "normal" (normal mode). The default mode is normal mode. If this node does not exist, the default mode is normal mode--></showMode>
    <menuTimeout min="" max=""><!--dependent, xs:integer, timeout period to exit, unit: second--></menuTimeout>
</IdentityTerminal>

```

## F.189 XML\_Cap\_ReaderAcrossHost

ReaderAcrossHost capability message in XML format

```

<ReaderAcrossHost version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <AcrossHostList size="8">
        <AcrossHostAction>
            <readerNo min="1" max="8"><!--req, xs: integer, card reader No.--></readerNo>
            <submarineBackEnabled opt="true,false">
                <!--req, xs: boolean, whether to enable the cross-controller anti-passing back function of the card reader-->
            </submarineBackEnabled>
        </AcrossHostAction>
    </AcrossHostList>
</ReaderAcrossHost>

```

```
</AcrossHostList>  
</ReaderAcrossHost>
```

### F.190 XML\_Cap\_ServerDevice

ServerDevice capability message in XML format

```
<ServerDevice version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <ipAddr min="" max=""><!--req, xs: string, IP address of the cross-controller  
  anti-passing back server--></ipAddr>  
  <port min="" max=""><!--req, xs: string, port No. of the cross-controller  
  anti-passing back server--></port>  
</ServerDevice>
```

### F.191 XML\_Cap\_StartReaderInfo

StartReaderInfo capability message in XML format

```
<StartReaderInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <hostNo min="1" max="64"><!--req, xs: integer, access controller No.--></  
  hostNo>  
  <readerNo min="1" max="8"><!--req, xs: integer, card reader No.--></readerNo>  
</StartReaderInfo>
```

### F.192 XML\_Cap\_SubmarineBack

SubmarineBack capability message in XML format

```
<SubmarineBack version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
  <enabled opt="true,false"><!--req, xs: boolean, whether to specify this  
  access controller as the cross-controller anti-passing back server--></enabled>  
</SubmarineBack>
```

### F.193 XML\_Cap\_SubmarineBackHostInfo

SubmarineBackHostInfo capability message in XML format

```
<SubmarineBackHostInfo version="2.0" xmlns="http://www.isapi.org/ver20/  
  XMLSchema">  
  <ID min="1" max="4"><!--req, xs: integer, configuration No.--></ID>  
  <HostInfoList size="16">  
    <Action>  
      <deviceNo min="1" max="64"><!--req, xs: integer, device No.--></deviceNo>  
      <serial min="9" max="9"><!--req, xs: string, device serial No.--></serial>  
    </Action>
```

```
</HostInfoList>  
</SubmarineBackHostInfo>
```

### F.194 XML\_Cap\_SubmarineBackMode

SubmarineBackMode capability message in XML format

```
<SubmarineBackMode version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
    <mode opt="disable,internetCommunicate,cardReadAndWrite"><!--req, xs:string,  
    anti-passing back mode--></mode>  
    <rule opt="line,inOrOut">  
        <!--req, xs:string, anti-passing back rule, this node is invalid when the  
        mode is set to "disable"-->  
    </rule>  
    <sectionID min="1" max="100">  
        <!--req, xs:integer, section ID, this node is valid when mode is  
        "cardReadAndWrite", and only one section ID can be configured for one  
        configuration-->  
    </sectionID>  
</SubmarineBackMode>
```

### F.195 XML\_Cap\_SubmarineBackReader

SubmarineBackReader capability message in XML format

```
<SubmarineBackReader version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  
    <ID min="1" max="128"><!--req, xs:integer, configuration No.--></ID>  
    <selfHostNo min="1" max="64"><!--req, xs:integer, access control No. of the  
    configuration object--></selfHostNo>  
    <selfReaderNo min="1" max="8"><!--req, xs:integer, card reader No. of the  
    configuration object--></selfReaderNo>  
    <FollowReaderList size="16">  
        <Action>  
            <followHostNo min="1" max="64"><!--req, xs:integer, following access  
            controller No.--></followHostNo>  
            <followReaderNo min="1" max="8"><!--req, xs:integer, following card  
            reader No.--></followReaderNo>  
        </Action>  
    </FollowReaderList>  
</SubmarineBackReader>
```

### F.196 XML\_ChannelControllerCfg

XML message about lane controller parameters

```
<ChannelControllerCfg version="2.0" xmlns="http://www.isapi.org/ver20/  
    XMLSchema">
```

```

<gatePassingMode><!--required, xs:string, turnstile passing mode:  

"ByChannelController"-based on the lane controller's local DIP settings,  

"ByRightController"-based on the main controller's settings--></gatePassingMode>  

    <freePassAuthEnabled><!--required, xs:string, whether the authentication is  

required for free passing: "enable"-yes, "disable"-no--></freePassAuthEnabled>  

    <openAndCloseSpeed><!--required, xs:integer, barrier's opening and closing  

speed, it is between 1 and 10, which represents the speed from 10% to 100%--></  

openAndCloseSpeed>  

    <alarmSoundTime><!--required, xs:integer, alarm prompt sound duration, unit:  

second. The value is between 0 and 599, and 0 refers to continuously playing  

alarm prompt sound--></alarmSoundTime>  

    <tempUnit><!--required, xs:string, temperature unit to be displayed:  

"Centigrade"-Celsius (°C), "Fahrenheit"-Fahrenheit (°F)--></tempUnit>  

    <alarmAreaNoAuth><!--optional, xs:boolean, whether opening door is prohibited  

in the alarm area--></alarmAreaNoAuth>  

    <gateWingMaterial><!--optional, xs:string, barrier material: "Acrylic"-  

acrylic, "StellPipe"-steel pipe, "SinglePUGate"-single PU gate, "DoublePUGate"-  

two PU gates--></gateWingMaterial>  

    <channelLength><!--optional, xs:integer, barrier length, unit: mm--></  

channelLength>  

    <motorDirection><!--optional, xs:string, motor rotation direction:  

"Clockwise", "AntiClockwise"--></motorDirection>  

    <lampBoardLight><!--optional, xs:integer, light board brightness, it is  

between 0 and 100--></lampBoardLight>  

    <openSpeed><!--optional, xs:string, barrier's opening speed, it is between 1  

and 10 which represents the speed from 10% to 100%, and the default speed is  

50%. If openAndCloseSpeed and openSpeed are both configured, the barrier's  

opening speed is determined by openSpeed--></openSpeed>  

    <closeSpeed><!--optional, xs:integer, barrier's closing speed, it is between  

1 and 10 which represents the speed from 10% to 100%, and the default speed is  

40%. If openAndCloseSpeed and closeSpeed are both configured, the barrier's  

closing speed is determined by closeSpeed--></closeSpeed>  

    <runMode><!--optional, xs:string, running mode: "doubleGateWing"-two barriers  

mode (default), "singleGateWing"-single barrier mode--></runMode>  

</ChannelControllerCfg>

```

## F.197 XML\_ClearCardRecord

ClearCardRecord message in XML format

```

<ClearCardRecord version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">  

    <clearAllCard>  

        <!--req, xs: boolean, whether to clear all card swiping records in the  

cross-controller anti-passing back server: "true"-yes, "false"-no. If this node  

is set to "false", either CardList or EmployeeNoList is required. If CardList  

is configured, it indicates clearing card swiping records by card No.; if  

EmployeeNoList is configured, it indicates clearing card swiping records by  

employee No.-->  

    </clearAllCard>  

    <CardList size="32">

```

```
<cardNo><!--opt, xs: string, card No., min="1" max="32"--></cardNo>
</CardList>
<EmployeeNoList size="32">
    <employeeNo><!--opt, xs:string, employee No. (person ID)--></employeeNo>
    </EmployeeNoList>
</ClearCardRecord>
```

### F.198 XML\_ClearSubmarineBack

ClearSubmarineBack message in XML format

```
<ClearSubmarineBack version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <clearHostInfo><!--opt, xs: boolean, whether to clear access controller
information: "true,false"--></clearHostInfo>
    <clearReaderInfo><!--opt, xs: boolean, whether to clear card reader
information: "true,false"--></clearReaderInfo>
    <clearSubmarineBack><!--opt, xs: boolean, whether to clear anti-passing back
server parameters: "true,false"--></clearSubmarineBack>
    <clearSubmarineBackHostInfo>
        <!--opt, xs: boolean, whether to clear cross-controller anti-passing back
parameters of access controllers: "true,false"-->
    </clearSubmarineBackHostInfo>
    <clearStartReaderInfo><!--opt, xs: boolean, whether to clear first card
reader parameters: "true,false"--></clearStartReaderInfo>
    <clearSubmarineBackReader>
        <!--opt, xs: boolean, whether to clear cross-controller anti-passing back
parameters of card readers: "true,false"-->
    </clearSubmarineBackReader>
    <clearSubmarineBackMode>
        <!--opt, xs: boolean, whether to clear the cross-controller anti-passing
back mode parameters: "true,false"-->
    </clearSubmarineBackMode>
    <clearServerDevice>
        <!--opt, xs: boolean, whether to clear the parameters of cross-controller
anti-passing back server: "true,false"-->
    </clearServerDevice>
    <clearReaderAcrossHost>
        <!--opt, xs: boolean, whether to clear the cross-controller anti-passing
back status of card readers: "true,false"-->
    </clearReaderAcrossHost>
</ClearSubmarineBack>
```

### F.199 XML\_DeployInfo

DeployInfo message in XML format

```
<DeployInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <DeployList size="5">
        <Content>
```

```

<deployNo><!--req, xs: integer, arming No.--></deployNo>
<deployType><!--req, xs: integer, arming type: 0-client arming to receive
real-time or offline events via platform or system (based on private protocol),
1-real-time arming to receive real-time events (based on private protocol), 2-
arm based on ISAPI protocol, opt="0,1,2"--></deployType>
<ipAddr><!--req, xs: string, IP address--></ipAddr>
</Content>
</DeployList>
</DeployInfo>

```

## F.200 XML\_DeviceCap

XML message about device capability

```

<DeviceCap version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
<SysCap><!--optional-->
    <isSupportDst><!--optional, xs: boolean, whether it supports daylight
saving time--></isSupportDst>
    <NetworkCap/><!--optional, xs: boolean, network capability-->
    <IOCap/><!--optional, IO capability-->
    <SerialCap/><!--optional, serial port capability-->
    <VideoCap/><!--optional, video capability, see details in the message of
XML_VideoCap-->
    <AudioCap/><!--optional, audio capability-->
    <isSupportHolidy><!--optional, xs:boolean--></isSupportHolidy>
    <RebootConfigurationCap>
        <Genetec><!--optional, xs:boolean--></Genetec>
        <ONVIF><!--optional, xs:boolean--></ONVIF>
        <RTSP><!--optional, xs:boolean--></RTSP>
        <HTTP><!--optional, xs:boolean--></HTTP>
        <SADP>
            <ISDiscoveryMode><!--optional, xs:boolean--></ISDiscoveryMode>
            <PcapMode><!--optional, xs:boolean--></PcapMode>
        </SADP>
        <IPCAAddStatus><!--optional, xs:boolean--></IPCAAddStatus>
    </RebootConfigurationCap>
    <isSupportExternalDevice><!--optional, xs:boolean--></
isSupportExternalDevice>
    <isSupportChangedUpload>
        <!--optional, xs: boolean, whether it supports uploading status changes-->
    </isSupportChangedUpload>
    <isSupportGettingWorkingStatus>
        <!--optional, xs:boolean, whether it supports getting device status-->
    </isSupportGettingWorkingStatus>
    <isSupportGettingChannelInfoByCondition>
        <!--optional, xs:boolean-->
    </isSupportGettingChannelInfoByCondition>
    <isSupportDiagnosedDataParameter>
        <!--optional, xs:boolean-->
    </isSupportDiagnosedDataParameter>

```

```

<isSupportSimpleDevStatus>
    <!--optional, xs: boolean, whether it supports getting device working
status-->
</isSupportSimpleDevStatus>
<isSupportFlexible>
    <!--optional, xs: boolean, whether it supports getting channel status by
condition-->
</isSupportFlexible>
<isSupportPTZChannels>
    <!--optional, xs:boolean, whether it supports returning PTZ channel
(which is different from the video channel)-->
</isSupportPTZChannels>
<isSupportSubscribeEvent>
    <!--optional, xs:boolean, whether it supports alarm or event
subscription: "true,false"-->
</isSupportSubscribeEvent>
<isSupportDiagnosedData>
    <!--optional, xs:boolean, "true,false", whether it supports diagnosis
data-->
</isSupportDiagnosedData>
<isSupportTimeCap>
    <!--optional, xs:boolean, whether it supports time capability-->
</isSupportTimeCap>
<isSupportThermalStreamData>
    <!--optional, xs:boolean, whether it supports uploading thermal stream
data in real-time. If it is supported, the returned value is "true"; otherwise,
this node will not be returned-->
</isSupportThermalStreamData>
<isSupportPostUpdateFirmware>
    <!--optional, xs:boolean, "true,false", whether it supports upgrading the
firmware-->
</isSupportPostUpdateFirmware>
<isSupportPostConfigData>
    <!--optional, xs:boolean, "true,false", whether it supports importing or
exporting the configuration file-->
</isSupportPostConfigData>
<isSupportUserLock>
    <!--optional, xs:boolean, "true,false", whether it supports locking user-->
</isSupportUserLock>
<isSupportModuleLock><!--optional, xs:boolean, whether it supports locking
the module: "true,false"--></isSupportModuleLock>
<isSupportSoundCfg><!--optional, xs:boolean--></isSupportSoundCfg>
<isSupportMetadata>
    <!--optional, xs:boolean, if it is supported, return "true", otherwise,
this node will not be returned-->
</isSupportMetadata>
<isSupportShutdown><!--optional, xs:boolean, whether it supports shutdown
configuration--></isSupportShutdown>
<supportSmartOverlapChannles opt="1"/><!--optional, xs:boolean, whether it
supports stream configuration of smart events. If this function is supported,
this node and the corresponding channel ID will be returned; otherwise, this
node will not be returned-->

```

```
<isSupportConsumptionMode><!--optional, xs:boolean, whether it supports  
switching power consumption mode:true (yes), this node is not returned (no).  
Related URI: /ISAPI/System/consumptionMode/capabilities?format=json--></  
isSupportConsumptionMode>  
    <isSupportManualPowerConsumption><!--optional, xs:boolean, whether it  
supports control the power consumption mode manually: true (yes), this node is  
not returned (no)--></isSupportManualPowerConsumption>  
    </SysCap>  
    <voicetalkNums><!--optional, xs:integer, the number of two-way audio  
channels--></voicetalkNums>  
    <isSupportSnapshot><!--optional, xs:boolean, whether it supports capture:  
"true, false"--></isSupportSnapshot>  
    <SecurityCap/><!--optional, security capability-->  
    <EventCap/><!--optional, event capability-->  
    <ITCCap><!--optional--></ITCCap>  
    <ImageCap/><!--optional, image capability-->  
    <RacmCap/><!--optional, storage capability-->  
    <PTZCtrlCap>  
        <isSupportPatrols><!--optional, xs:boolean--></isSupportPatrols>  
        <isSupportCombinedPath><!--optional, xs:boolean, whether the device  
supports the PTZ combined path-->true</isSupportCombinedPath>  
    </PTZCtrlCap>  
    <SmartCap/><!--optional, intelligent capability-->  
    <isSupportEhome><!--optional, xs:boolean--></isSupportEhome>  
    <isSupportStreamingEncrypt><!--optional, xs:boolean--></  
isSupportStreamingEncrypt>  
    <TestCap>  
        <isSupportEmailTest><!--optional, xs:boolean--></isSupportEmailTest>  
    </TestCap>  
    <ThermalCap/><!--optional, temperature measurement capability-->  
    <WLAlarmCap/><!--optional, wireless alarm capability-->  
    <SecurityCPCapabilities/><!--optional, security control panel capability-->  
    <isSupportGIS>  
        <!--optional, xs:boolean, whether it supports GIS capability-->  
    </isSupportGIS>  
    <isSupportCompass>  
        <!--optional, xs:boolean-->  
    </isSupportCompass>  
    <isSupportRoadInfoOverlays>  
        <!--optional, xs:boolean-->  
    </isSupportRoadInfoOverlays>  
    <isSupportFaceCaptureStatistics>  
        <!--optional, xs:boolean-->  
    </isSupportFaceCaptureStatistics>  
    <isSupportExternalDevice>  
        <!--optional, xs:boolean-->  
    </isSupportExternalDevice>  
    <isSupportElectronicsEnlarge>  
        <!--optional, xs:boolean, whether it supports digital zoom-->  
    </isSupportElectronicsEnlarge>  
    <isSupportRemoveStorage>  
        <!--optional, xs:boolean-->
```

```

</isSupportRemoveStorage>
<isSupportCloud>
    <!--optional, xs:boolean-->
</isSupportCloud>
<isSupportRecordHost>
    <!--optional, xs:boolean-->
</isSupportRecordHost>
<isSupportEagleEye>
    <!--optional, xs:boolean, whether it supports PanoVu series camera-->
</isSupportEagleEye>
<isSupportPanorama>
    <!--optional, xs:boolean, whether it supports panorama-->
</isSupportPanorama>
<isSupportFirmwareVersionInfo>
    <!--optional, xs:boolean, whether it supports displaying firmware version
information-->
</isSupportFirmwareVersionInfo>
<isSupportExternalWirelessServer>
    <!--optional, xs: boolean-->
</isSupportExternalWirelessServer>
<isSupportSetupCalibration>
    <!--optional, xs:boolean, whether it supports setting calibration-->
</isSupportSetupCalibration>
<isSupportGetmutexFuncErrMsg>
    <!--optional, xs:boolean, whether it supports getting mutex information-->
</isSupportGetmutexFuncErrMsg>
<isSupportTokenAuthenticate><!--optional, xs:boolean--></
isSupportTokenAuthenticate>
<isSupportStreamDualVCA><!--optional, xs:boolean--></isSupportStreamDualVCA>
<isSupportlaserSpotManual>
    <!--optional, boolean, whether it supports laser spot configuration-->
</isSupportlaserSpotManual>
<isSupportRTMP><!--optional, xs:boolean--></isSupportRTMP>
<isSupportTraffic><!--optional, xs:boolean--></isSupportTraffic>
<isSupportLaserSpotAdjustment>
    <!--optional, boolean, whether it supports adjusting laser spot size-->
</isSupportLaserSpotAdjustment>
<VideoIntercomCap/><!--optional, video intercom capability-->
<isSupportSafetyCabin>
    <!--optional, xs:boolean-->
</isSupportSafetyCabin>
<isSupportPEA>
    <!--optional, xs:boolean, whether it supports one-touch security control
panel capability-->
</isSupportPEA>
<isSupportCurrentLock>
    <!--optional, xs:boolean, whether it supports locking current
configuration-->
</isSupportCurrentLock>
<isSupportGuardAgainstTheft>
    <!--optional, xs:boolean, whether it supports device anti-theft
configuration-->

```

```

</isSupportGuardAgainstTheft>
<isSupportPicInfoOverlap>
    <!--optional, xs:boolean, whether it supports picture information overlay-->
</isSupportPicInfoOverlap>
<isSupportPlay>
    <!--optional, xs: boolean, whether it supports live view: "true,false"-->
</isSupportPlay>
<isSupportPlayback>
    <!--optional, xs: boolean, whether it supports playback: "true,false"-->
</isSupportPlayback>
<UHFRFIDReader>
    <!--optional, supported capability of UHF RFID card reader-->
    <isSupportBasicInformation>
        <!--optional, xs:boolean, whether it supports basic parameters of UHF
RFID card reader-->
    </isSupportBasicInformation>
    <isSupportHardDiskStorageTest>
        <!--optional, xs:boolean, whether it supports hard disk storage test of
UHF RFID card reader-->
    </isSupportHardDiskStorageTest>
</UHFRFIDReader>
<isSupportIntelligentStructureAnalysis>
    <!--optional, xs:boolean, whether it supports structured VCA-->
</isSupportIntelligentStructureAnalysis>
<isSupportIntelligentAnalysisEngines>
    <!--optional, xs:boolean, whether it supports VCA engine configuration-->
</isSupportIntelligentAnalysisEngines>
<PreviewDisplayNum>
    <!--optional, xs:integer, the number of live view windows, which is the
number of simultaneous live view windows controlled by the device. Limited by
the performance of DeepinMind series network video recorder, currently only
live view of a network camera is supported, and playback is not supported-->
</PreviewDisplayNum>
<isSupportBoard opt="true,false">
    <!--optional, xs:boolean, whether it supports protocol related to sub-
board-->
</isSupportBoard>
<ResourceSwitch>
    <workMode opt="4KPreview,educationRecord">
        <!--req, xs:string, device working mode: "4KPreview"-4K live view mode,
"educationRecord"-education recording mode-->
    </workMode>
</ResourceSwitch>
<isSupportCustomStream><!--optional, xs:boolean--></isSupportCustomStream>
<isSupportTriggerCapCheck>
    <!--optional, xs:boolean, whether it supports verifying capability of alarm
linkage actions-->
</isSupportTriggerCapCheck>
<isSupportActiveMulticast>
    <!--optional, xs: boolean, whether it supports active multicast-->
</isSupportActiveMulticast>
<isSupportChannelEventCap>

```

```

<!--optional, xs:boolean, whether it supports getting event capability by
channel-->
</isSupportChannelEventCap>
<isSupportPictureServer>
    <!-- opt, xs:boolean, whether it supports picture storage server-->
</isSupportPictureServer>
<isSupportVideoCompositeAlarm>
    <!--optional, xs:boolean, whether it supports video double check alarm-->
</isSupportVideoCompositeAlarm>
<isSupportSensorCalibrating>
    <!--optional, xs:boolean, whether it supports double sensor calibration-->
</isSupportSensorCalibrating>
<isSupportChannelEventListCap>
    <!--optional, xs:boolean, whether it supports getting event capability of
all channels-->
</isSupportChannelEventListCap>
<VCAResourceChannelsCap>
    <!--optional, whether it supports independently switching to another VCA
resource by channel-->
    <ChannelsList>
        <channelsID>
            <!--req, xs:integer, channel No. supported by the device-->
        </channelsID>
    </ChannelsList>
</VCAResourceChannelsCap>
<SensorCap/><!--optional, intelligent cabinet capability-->
<isSupportSecurityCP/>
    <!--optional, xs:boolean, whether it supports the applications of security
control panel: "true, false"-->
</isSupportSecurityCP>
<isSupportClientProxyWEB>
    <!--optional, xs:boolean, whether it supports the function that the client
proxy passes through the remote web configuration: "true"-->
</isSupportClientProxyWEB>
<WEBLocation>
    <!--optional, string type, web page location: "local"-local device,
"remote"-remote location. If this node is not returned, the web page will be in
the local device by default-->
</WEBLocation>
<isSupportTime/>
    <!--optional, xs:boolean, "true, false", whether it supports time
configuration-->
</isSupportTime>
<isSupportTimeZone/>
    <!--optional, xs:boolean, "true, false", whether it supports daylight
saving time (DST) configuration-->
</isSupportTimeZone>
<isSupportCityManagement>
    <!--optional, boolean, ro, whether it supports intelligent city management-->
</isSupportCityManagement>
<isSupportMixedTargetDetection>

```

```

<!--optional, xs:boolean, "true, false", whether it supports multi-target-
type detection-->
</isSupportMixedTargetDetection>
<isSupportFaceContrastMode>
    <!--optional, xs:boolean, whether it supports face picture comparison mode-->
</isSupportFaceContrastMode>
<isSupportPictureCaptureComparision>
    <!--optional, xs:boolean, whether it supports face picture N:1 comparison
between face pictures captured by the camera and imported face pictures-->
</isSupportPictureCaptureComparision>
<isSupportGPSCalibratation>
    <!--optional, xs:boolean, whether it supports GPS calibration capability-->
</isSupportGPSCalibratation>
<isSupportChannelFullEventListCap>
    <!--optional, xs:boolean, whether it supports getting event list capability
of all channels-->
</isSupportChannelFullEventListCap>
<isSupportAUXInfoCap>
    <!--optional, xs:boolean, whether it supports getting property capability
of all channels-->
</isSupportAUXInfoCap>
<isSupportCalibrationFile>
    <!--optional, xs:boolean, whether it supports importing calibration file-->
</isSupportCalibrationFile>
<isSupportDisplayTrajectory>
    <!--optional, xs:boolean, whether it supports displaying trajectory-->
</isSupportDisplayTrajectory>
<maximumSuperPositionTime opt="5,10,20,30">
    <!--dep, xs:integer, the maximum time of trajectory displaying, unit:
second, it is valid only when displaying trajectory is supported-->
</maximumSuperPositionTime>
<isSupportUnitConfig>
    <!--optional, xs:boolean, whether it supports unit configuration-->
</isSupportUnitConfig>
<isSupportAutoMaintenance>
    <!--optional, xs:boolean, whether it supports automatic maintenance. When
this node exists and values "true", it indicates support-->
</isSupportAutoMaintenance>
<isSupportGetLinkSocketIP>
    <!--optional, xs: boolean, "true,false", whether it supports getting the
SocketIP of current connection-->
</isSupportGetLinkSocketIP>
<isSupportIntelligentSearch>
    <!--optional, xs:boolean, whether it supports intelligent search-->
</isSupportIntelligentSearch>
<IOTCap><!--optional, xs:boolean, IoT device access capability-->
    <supportChannelNum>
        <!--req, xs:integer, number of supported channels of IoT device-->
    </supportChannelNum>
    <startChannelNo>
        <!--optional, xs:integer, initial channel ID, if this node is not

```

```

inputted, it indicates that the initial channel ID is 1-->
    </startChannelNo>
    <isSupportlinkageChannelsSearch>
        <!--optional, boolean, returns "true" if support, returns "false" if not
support-->
        </isSupportlinkageChannelsSearch>
    </IOTCap>
    <isSupportEncryption>
        <!--optional, xs: boolean, stream encryption capability-->
    </isSupportEncryption>
    <AIDEEventSupport opt="abandonedObject, pedestrian, congestion, roadBlock,
construction, trafficAccident, fogDetection, wrongDirection, illegalParking,
SSharpDriving, lowSpeed, dragRacing">
        <!--optional, xs:string, supported traffic incident type: "abandonedObject"-->
objects dropped down, "pedestrian"-pedestrian, "congestion"-congestion,
"roadBlock"-roadblock, "construction"-construction, "trafficAccident"-traffic
accident, "fogDetection"-fog, "wrongDirection"-wrong-way driving,
"illegalParking"-illegal parking, "SSharpDriving"-slalom driving, "lowSpeed"->
        </AIDEEventSupport>
        <TFSEventSupport
opt="illegalParking ,wrongDirection,crossLane,laneChange,vehicleExist,turnRound,
parallelParking,notKeepDistance,notSlowZebraCrossing,overtakeRightSide,lowSpeed,
dragRacing,changeLaneContinuously,SSharpDriving,largeVehicleOccupyLine,jamCrossL
ine">
            <!--optional, xs:string, supported enforcement event type: "illegalParking"-->
illegal parking, "wrongDirection"-wrong-way driving, "crossLane"-driving on the
lane line, "laneChange"-illegal lane change, "vehicleExist"-motor vehicle on
non-motor vehicle lane, "turnRound"-illegal U-turn, "parallelParking"-parallel
parking, "notKeepDistance"-not keeping vehicle distance, "notSlowZebraCrossing"-->
not slowing down at zebra corssing, "overtakeRightSide"-overtaking on the
right, "lowSpeed"-driving in low speed, "dragRacing"-street racing,
"changeLaneContinuously"-continuous lane change, "SSharpDriving"-slalom
driving, "largeVehicleOccupyLine"-lane occupation by large-sized vehicle,
"jamCrossLine"-queue jumping-->
        </TFSEventSupport>
        <isVehicleStatisticsSupport>
            <!--optional, xs: boolean, whether it supports setting parameters for
traffic data collection-->
        </isVehicleStatisticsSupport>
        <isSupportIntersectionAnalysis>
            <!--optional, xs: boolean, whether it supports intersection analysis-->
        </isSupportIntersectionAnalysis>
        <supportRemoteCtrl
opt="up,down,left,right,enter,menu,num,power,esc,edit,F1,.prev,rec,play,stop,not
Support"/><!--whether it supports remote control-->
        <isSptDiagnosis>
            <!--optional, xs:boolean, whether it supports device diagnosis: "true",
"false"-->
        </isSptDiagnosis>
        <isSptSerialLogCfg>
            <!--optional, xs:boolean, whether it supports configuring serial port log

```

```

 redirection: "true", "false"-->
 </isSptSerialLogCfg>
 <isSptFileExport>
    <!--optional, xs:boolean, whether it supports exporting files from the
device: "true", "false"-->
 </isSptFileExport>
 <isSptCertificationStandard>
    <!--optional, xs:boolean, whether it supports configuring authentication
standard for security control panel: "true", "false"-->
 </isSptCertificationStandard>
 <isSptKeypadLock>
    <!--optional, xs:boolean, whether it supports locking keypad: "true",
"false"-->
 </isSptKeypadLock>
 <MixedTargetDetection><!--optional, whether the device supports recognizing
specific target among mixed targets-->
    <isSupportFaceRecognition><!--optional, xs:boolean, whether it supports
face recognition--></isSupportFaceRecognition>
    <isSupportHumanRecognition><!--optional, xs:boolean, whether it supports
human body recognition--></isSupportHumanRecognition>
    <isSupportVehicleRecognition><!--optional, xs:boolean, whether it supports
vehicle recognition--></isSupportVehicleRecognition>
 </MixedTargetDetection>
 <isSupportDiscoveryMode><!--optional, xs:boolean--></isSupportDiscoveryMode>
 <streamEncryptionType>
    <!--dep, xs:string, stream encryption type: "RTP/TLS", "SRTP/UDP", "SRTP/
MULTICAST". This node is valid when <isSupportEncryption> is "true", and the
device can support one or more stream encryption types-->
 </streamEncryptionType>
 <isSupportLms><!--optional, xs:boolean, whether it supports laser--></
isSupportLms>
 <isSupportLCDScreen><!--optional, xs:boolean, whether it supports LCD screen-->
 </isSupportLCDScreen>
 <isSupportBluetooth><!--optional, xs:boolean, whether it supports bluetooth-->
 </isSupportBluetooth>
 <isSupportAcsUpdate>
    <!--optional, whether it supports upgrading sub access control devices or
peripheral modules: "true"-yes, this node is not returned-no-->
 </isSupportAcsUpdate>
 <isSupportAccessControlCap>
    <!--optional, whether it supports access control capability: "true"-yes,
this node is not returned-no-->
 </isSupportAccessControlCap>
 <isSupportIDCardInfoEvent><!--optional, whether it supports ID card swiping
event: "true"-yes. This node will not be returned if this function is not
supported--></isSupportIDCardInfoEvent>
 <OpenPlatformCap><!--optional, embedded open platform capability, refer to
the message XML_OpenPlatformCap for details-->
 <isSupportInstallationAngleCalibration>
    <!--optional, xs:boolean, whether it supports installation angle
calibration-->
 </isSupportInstallationAngleCalibration>

```

```

<isSupportZeroBiasCalibration>
    <!--optional, xs:boolean, whether it supports zero bias calibration-->
</isSupportZeroBiasCalibration>
<isSupportDevStatus><!--optional, xs:boolean, whether device supports getting
device status--></isSupportDevStatus>
<isSupportRadar><!--optional, xs:boolean, whether it supports the security
radar--></isSupportRadar>
<isSupportRadarChannels><!--optional, xs:boolean, whether it supports getting
radar channels--></isSupportRadarChannels>
<radarIPDForm><!--optional, xs:string, radar form: "single"-single radar,
"double_diagonal"-two radars forming an 180° diagonal, "double_vertical"-two
radars forming a 90° vertical angle--></radarIPDForm>
<isSupportRadarFieldDetection><!--optional, xs:boolean, whether it supports
intrusion detection (radar)--></isSupportRadarFieldDetection>
<isSupportRadarLineDetection><!--optional, xs:boolean, whether it supports
line crossing detection (radar)--></isSupportRadarLineDetection>
<mixedTargetDetectionWebNoDisplay><!--optional, xs:boolean, whether to enable
not displaying multi-target-type recognition--><
mixedTargetDetectionWebNoDisplay>
<SHMCap><!--opt-->
    <isSupportHighHDTemperature><!--optional, xs:boolean, whether it supports
HDD high temperature detection--></isSupportHighHDTemperature>
    <isSupportLowHDTemperature><!--optional, xs:boolean, whether it supports
HDD low temperature detection--></isSupportLowHDTemperature>
    <isSupportHDImpact><!--optional, xs:boolean, whether it supports HDD impact
detection--></isSupportHDImpact>
    <isSupportHDBadBlock><!--optional, xs:boolean, whether it supports HDD bad
sector detection--></isSupportHDBadBlock>
    <isSupportSevereHDFailure><!--optional, xs:boolean, whether it supports HDD
severe fault detection--></isSupportSevereHDFailure>
</SHMCap>
<isSupportBVCorrect><!--optional, xs:boolean, whether it supports configuring
camera correction parameters--></isSupportBVCorrect>
<guideEventSupport opt="linkageCapture">
    <!--optional,xs:string, events which support quick setup by instruction,
"linkageCapture"-capture by linkage-->
</guideEventSupport>
<isSupportAutoSwitch><!--optional, xs:boolean, whether it supports auto
switch--> true</isSupportAutoSwitch>
<isSupportDataPrealarm><!--optional,xs:boolean, whether it supports traffic
pre-alarm event--></isSupportDataPrealarm>
<supportGISEvent opt="AID,TPS,ANPR,mixedTargetDetection">
    <!--optional, xs:string, event types that support GIS information access:
AID (corresponding SDK event: COMM_ALARM_AID_V41), TPS (corresponding SDK
event: COMM_ALARM_TPS_REAL_TIME), ANPR (corresponding SDK event:
COMM_ITS_PLATE_RESULT), mixedTargetDetection-mixed targets detection-->
</supportGISEvent>
<isSupportIntelligentMode><!--optional, xs:boolean, whether it supports
intelligent scene switch (related URI:/ISAPI/System/IntelligentSceneSwitch?
format=json)--></isSupportIntelligentMode>
<isSupportCertificateCaptureEvent><!--optional, xs:boolean, whether it
supports certificate capture and comparison events: true-yes. If this function

```

```

is not supported, this node will not be returned--></
isSupportCertificateCaptureEvent>
<isSupportAlgorithmsInfo><!--optional, xs:boolean, whether it supports
getting the algorithm library version information: true-yes. If this function
is not supported, this node will not be returned--></isSupportAlgorithmsInfo>
<isSupportVibrationDetection><!--optional, xs:boolean, whether it supports
vibration detection--></isSupportVibrationDetection>
<isSupportFaceTemperatureMeasurementEvent><!--optional, xs:boolean, whether
it supports uploading face thermography events (eventType:
"FaceTemperatureMeasurementEvent")--></isSupportFaceTemperatureMeasurementEvent>
<isSupportQRCodeEvent><!--optional, xs:boolean, whether it supports uploading
QR code events (eventType: "QRCodeEvent")--></isSupportQRCodeEvent>
<isSupportPersonArmingTrack><!--optional, xs:boolean, whether device supports
person arming (related URI: /ISAPI/Intelligent/channels/<ID>/personArmingTrack/
capabilities?format=json)--></isSupportPersonArmingTrack>
<isSupportManualPersonArmingTrack><!--optional, xs:boolean, whether device
supports manual person arming (related URI: /ISAPI/Intelligent/channels/<ID>/
manualPersonArmingTrack?format=json)--></isSupportManualPersonArmingTrack>
<isSupportGPSCalibrationMode><!--optional, xs:boolean, whether device
supports GPS calibration (related URI: /ISAPI/System/GPSCalibration/channels/
<ID>/mode?format=json)--></isSupportGPSCalibrationMode>
<isSupportGPSVerification><!--optional, xs:boolean, whether device supports
GPS verification (related URI: /ISAPI/System/GPSVerification/channels/<ID>/
points?format=json)--></isSupportGPSVerification>
<isSupportHBDLib><!--optional, xs:boolean, whether device supports human body
picture library (related URI: /ISAPI/Intelligent/HBDLib/capabilities?
format=json)--></isSupportHBDLib>
<isSupportFireEscapeDetection><!--optional, xs:boolean, whether the device
supports fire engine access detection (related URI: /ISAPI/Intelligent/channels/
<ID>/fireEscapeDetection/capabilities?format=json)--></
isSupportFireEscapeDetection>
<isSupportTakingElevatorDetection><!--optional, xs:boolean, whether the device
supports elevator detection (related URI: /ISAPI/Intelligent/channels/
<ID>/takingElevatorDetection/capabilities?format=json)--></
isSupportTakingElevatorDetection>
<isSupportSSDFileSystemUpgrade><!--optional, xs:boolean, whether the device
supports SSD file system upgrade (related URI: /ISAPI/System/SSDFileSystem/
upgrade?format=json)--></isSupportSSDFileSystemUpgrade>
<isSupportSSDFileSystemFormat><!--optional, xs:boolean, whether the device
supports SSD file system formatting (related URI: /ISAPI/System/SSDFileSystem/
format?format=json)--></isSupportSSDFileSystemFormat>
<isSupportSSDFileSystemCapacity><!--optional, xs:boolean, whether the device
supports getting space distribution information of SSD file system (related
URI: /ISAPI/System/SSDFileSystem/capacity?format=json)--></
isSupportSSDFileSystemCapacity>
<isSupportAIOpenPlatform><!--optional, xs:boolean, whether the device
supports AI open platform capabilities; if supports, this node will be returned
and its value is true; if not, this node will not be returned--></
isSupportAIOpenPlatform>
<isSupportPictureDownloadError><!--optional, xs:boolean, whether the device
supports reporting picture download failure--></isSupportPictureDownloadError>
<characteristicCode min="1" max="128"><!--optional, xs:string, device

```

```

attribute code (related URI: /ISAPI/System/deviceInfo/characteristicCode?
format=json)--></characteristicCode>
<isSupportContainerDetection><!--optional, xs:boolean, whether the device
supports container detection (if this node is not returned, refer to the value
returned by /ISAPI/Traffic/ContentMgmt/InputProxy/channels/<ID>/ocrScene/
capabilities to find whether the device supports container detection)--></
isSupportContainerDetection>
<isSupportLensParamFile><!--optional, xs:boolean, whether the device supports
exporting and importing the lens parameters file--></isSupportLensParamFile>
<isSupportCounting><!--optional, xs:boolean, ro, whether it supports people
counting--></isSupportCounting>
<isSupportFramesPeopleCounting><!--optional, xs:boolean, ro, whether it
supports regional people counting--></isSupportFramesPeopleCounting>
<zoomFocusWebDisplay
opt="ROI,roadTrafficDetection,SMD,mixedTargetDetection,faceCapture"><!--
optional, string, zoom and focus page supported by the Web Client--></
zoomFocusWebDisplay>
<isSupportDebugLogModuleType
opt="playService,communicationService,attendanceService,faceService"><!--
optional, xs:boolean, whether to export the debugging logs by module type; the
value of <moduleType> in the URI (/ISAPI/System.debugLineLog?
format=json&moduleType=<moduleType>) can be: "playService",
"communicationService", "attendanceService", "faceService"--></
isSupportDebugLogModuleType>
</isSupportPlateQuaAlarm>
<isSupportWiegand><!--optional, xs:boolean, ro, whether it supports the
Wiegand protocol (related URI: /ISAPI/System/Wiegand/<wiegandID>/capabilities?
format=json)-->true</isSupportWiegand>
<isSupportChannelOccupy><!--optional, xs:boolean, whether it supports
detection of outdoor fire escape occupied by vehicle--></isSupportChannelOccupy>
<isSupportOffDuty><!--optional, xs:boolean, whether it supports detection of
person absent in fire control room--></isSupportOffDuty>
<isSupportNoCertificate><!--optional, xs:boolean, whether it supports
detection of authenticated staff not enough in fire control room--></
isSupportNoCertificate>
<isSupportSmokeAlarm><!--optional, xs:boolean, whether it supports smoke
alarm--></isSupportSmokeAlarm>
<isSupportBatteryCarDisobey><!--optional, xs:boolean, whether it supports
electric scooter parking violation detection--></isSupportBatteryCarDisobey>
<isSupportNoFireExtinguisherRecog><!--optional, xs:boolean, whether it
supports fire extinguisher missing detection--></
isSupportNoFireExtinguisherRecog>
<isSupportIndoorPasswayBlock><!--optional, xs:boolean, whether it supports
indoor channel blockage detection--></isSupportIndoorPasswayBlock>
<isSupportFireSmartFireDetect><!--optional, xs:boolean, whether it supports
fire source detection--></isSupportFireSmartFireDetect>
<isSupportDetectorRunningStatus><!--optional, xs:boolean, whether it supports
detector running status--></isSupportDetectorRunningStatus>
<isSupportDetectorOperationStatus><!--optional, xs:boolean, whether it
supports detector operation status--></isSupportDetectorOperationStatus>
<isSupportDetectorTemperatureAlarm
opt="highTemperature,riseTemperature,flame"><!--optional, xs:boolean, whether

```

```
it supports temperature alarm: "highTemperature" (high temperature alarm),  
"riseTemperature" (temperature rising alarm), "flame" (flame alarm)--></  
isSupportDetectorTemperatureAlarm>  
    <isSupportDetectorShelterAlarm><!--optional, xs:boolean, whether it supports  
detector video tampering alarm--></isSupportDetectorShelterAlarm>  
    <isSupportDetectorMotionAlarm><!--optional, xs:boolean, whether it supports  
detector movement alarm--></isSupportDetectorMotionAlarm>  
    <isSupportDetectorTamperAlarm><!--optional, xs:boolean, whether it supports  
detector tampering alarm--></isSupportDetectorTamperAlarm>  
    <isSupportDetectorEmergencyAlarm><!--optional, xs:boolean, whether it  
supports detector emergency alarm--></isSupportDetectorEmergencyAlarm>  
    <isSupportSmokingDetectAlarm><!--optional, xs:boolean, whether it supports  
smoking alarm--></isSupportSmokingDetectAlarm>  
    <isSupportDetectorSmokeAlarm><!--optional, xs:boolean, whether it supports  
smoke alarm--></isSupportDetectorSmokeAlarm>  
    <isSupportDetectorCombustibleGasAlarm><!--optional, xs:boolean, whether it  
supports gas alarm--></isSupportDetectorCombustibleGasAlarm>  
    <isSupportFireControlData><!--optional, xs:boolean, whether it supports  
uploading real-time fire protection data--></isSupportFireControlData>  
    <isSupportFireNoRegulation><!--optional, xs:boolean, whether it supports fire  
no regulation alarm--></isSupportFireNoRegulation>  
    <isSupportSmokeFireRecognize><!--optional, xs:boolean, whether it supports  
uploading the smoke and fire detection event--></isSupportSmokeFireRecognize>  
</DeviceCap>
```

## F.201 XML\_Desc\_AcsAbility

Input description message for getting access control capability.

```
<AcsAbility version="2.0">  
    <!--opt, specify child nodes about access control capabilities to be  
returned-->  
</AcsAbility>
```

## F.202 XML\_EventNotificationAlert\_AlarmEventInfo

EventNotificationAlert message with alarm/event information in XML format.

```
<EventNotificationAlert version="2.0" xmlns="http://www.isapi.org/ver20/  
XMLSchema">  
    <ipAddress><!--dep, xs:string, device IPv4 address--></ipAddress>  
    <ipv6Address><!--dep, xs:string, device IPv6 address--></ipv6Address>  
    <portNo><!--opt, xs:integer, device port number--></portNo>  
    <protocol><!--opt, xs:string, protocol type for uploading alarm/event  
information, "HTTP,HTTPS"--></protocol>  
    <macAddress><!--opt, xs:string, MAC address--></macAddress>  
    <channelID><!--dep, xs:string, device channel No., starts from 1--></
```

```

channelID>
    <dateTime><!--req, alarm/event triggered or occurred time, format:
2017-07-19T10:06:41+08:00--></dateTime>
    <activePostCount><!--req, xs:integer, alarm/event frequency, starts from 1--
></activePostCount>
    <eventType><!--req, xs:string, alarm/event type, "peopleCounting, ANPR,..."--
></eventType>
    <eventState>
        <!--req, xs:string, durative alarm/event status: "active"-valid, "inactive"-invalid,
e.g., when a moving target is detected,
the alarm/event information will be uploaded continuously until the status
is set to "inactive"-->
    </eventState>
    <eventDescription><!--req, xs:string, alarm/event description--></
eventDescription>
    <...><!--opt, for different alarm/event types, the nodes are different, see
the message examples in different applications--></...>
</EventNotificationAlert>

```

### F.203 XML\_FaceCompareCond

XML message about condition parameters of face picture comparison

```

<FaceCompareCond version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <faceWidthLowerLimit><!--optional, xs:integer, face width threshold with
highest priority, value range: [0, 100], when the detected face width is larger
than this threshold, the following conditions will be ignored and the face
comparison will be executed--></faceWidthLowerLimit>
    <pitch><!--optional, xs:integer, face raising or bowing angle, value range:
[0, 90], unit: degree, the smaller the better--></pitch>
    <yaw><!--optional, xs:integer, face siding left or right angle, value range:
[0, 90], unit: degree, the smaller the better--></yaw>
    <width><!--optional, xs:integer, face width, value range: [0, 100]--></width>
    <height><!--optional, xs:integer, face height, value range: [0, 100]--></
height>
    <leftBorder><!--optional, xs:integer, left border of face, value range: [0,
100]--></leftBorder>
    <rightBorder><!--optional, xs:integer, right border of face, value range: [0,
100]--></rightBorder>
    <upBorder><!--optional, xs:integer, top border of face, value range: [0,
100]--></upBorder>
    <bottomBorder><!--optional, xs:integer, bottom border of face, value range:
[0, 100]--></bottomBorder>
    <interorbitalDistance><!--optional, xs:integer, pupil distance, value range:
[0, 100]--></interorbitalDistance>
    <faceScore><!--optional, xs:integer, face score, value range: [0, 100], the
valid face score must be larger than this score--></faceScore>
    <maxDistance><!--optional, xs:string, maximum recognition distance:
"0.5,1,1.5,2,auto", unit: m. This node has higher priority over
<interorbitalDistance>--></maxDistance>

```

```

<similarity><!--optional, xs:float, face comparison similarity, value range:
[0.0,1.0]--></similarity>
</FaceCompareCond>
```

### **F.204 XML\_IdentityTerminal**

IdentityTerminal message in XML format

```

<IdentityTerminal version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <terminalMode>
        <!--req, xs: string, terminal mode: "authMode"-authentication mode,
        "registerMode"-registration mode-->
    </terminalMode>
    <idCardReader>
        <!--req, xs: string, ID card reader model: iDR210, DS-K1F110-I, DS-K1F1110-
        B, DS-K1F1110-AB, none, DS-K1F1001-I(USB), DS-K1F1002-I(USB), none-->
    </idCardReader>
    <camera><!--req, xs: string, camera model: C270, DS-2CS5432B-S--></camera>
    <fingerPrintModule><!--req, xs: string, fingerprint module type: ALIWARD,
    HikModule--></fingerPrintModule>
    <videoStorageTime><!--req, xs: integer, time for saving video (unit: second),
    which is between 0 and 10--></videoStorageTime>
    <faceContrastThreshold><!--req, xs: integer, face picture comparison
    threshold, which is between 0 and 100--></faceContrastThreshold>
    <twoDimensionCode><!--req, xs: string, whether to enable QR code recognition:
    enable, disable--></twoDimensionCode>
    <blackListCheck><!--req, xs: string, whether to enable blocklist
    verification: enable, disable--></blackListCheck>
    <idCardCheckCenter>
        <!--req, xs: string, ID card comparison mode: local-compare with ID card of
        local storage, server-compare with ID card of remote server storage-->
    </idCardCheckCenter>
    <faceAlgorithm>
        <!--req, xs: string, face picture algorithm: HIK-Z-Private algorithm, HIK-
        third-party algorithm-->
    </faceAlgorithm>
    <comNo><!--req, xs: integer, COM No., which is between 1 and 9--></comNo>
    <memoryLearning><!--req, xs: string, whether to enable learning and memory
    function: enable, disable--></memoryLearning>
    <saveCertifiedImage><!--req, xs: string, whether to enable saving
    authenticated picture: enable, disable--></saveCertifiedImage>
    <MCUVersion><!--opt, xs: string, MCU version information, read-only--></
    MCUVersion>
    <usbOutput><!--opt, xs: string, whether to enable USB output of ID card
    reader: enable, disable--></usbOutput>
    <serialOutput><!--opt, xs: string, whether to enable serial port output of ID
    card reader: enable, disable--></serialOutput>
    <readInfoOfCard><!--opt, xs: string, set content to be read from CPU card:
    serialNo-read serial No., file-read file--></readInfoOfCard>
    <workMode><!--opt, xs: string, authentication mode: passMode,
```

```

accessControlMode--></workMode>
    <ecoMode>
        <eco><!--opt, xs: string, whether to enable ECO mode: enable, disable--></
eco>
            <faceMatchThreshold1><!--req, xs: integer, 1V1 face picture comparison
threshold of ECO mode, which is between 0 and 100--></faceMatchThreshold1>
            <faceMatchThresholdN><!--req, xs: integer, 1:N face picture comparison
threshold of ECO mode, which is between 0 and 100--></faceMatchThresholdN>
            <changeThreshold><!--opt, xs: string, switching threshold of ECO mode,
which is between 0 and 8--></changeThreshold>
            <maskFaceMatchThresholdN><!--req, xs:integer, 1:N face picture (face with
mask and normal background picture) comparison threshold of ECO mode, value
range: [0,100]--></maskFaceMatchThresholdN>
            <maskFaceMatchThreshold1><!--req, xs:integer, 1:1 face picture (face with
mask and normal background picture) comparison threshold of ECO mode, value
range: [0,100]--></maskFaceMatchThreshold1>
        </ecoMode>
        <readCardRule><!--opt, xs: string, card No. setting rule: "wiegand26",
"wiegand34"--></readCardRule>
        <enableScreenOff><!--optional, xs:boolean, whether the device enters the
sleep mode when there is no operation after the configured sleep time--></
enableScreenOff>
        <screenOffTimeout><!--dependent, xs:integer, sleep time, unit: second--></
screenOffTimeout>
        <enableScreensaver><!--optional, xs:boolean, whether to enable the screen
saver function--></enableScreensaver>
        <showMode><!--optional, xs:string, display mode: "concise" (simple mode, only
the authentication result will be displayed), "normal" (normal mode). The
default mode is normal mode. If this node does not exist, the default mode is
normal mode--></showMode>
        <menuTimeout><!--dependent, xs:integer, timeout period to exit, unit: second-->
    </menuTimeout>
</IdentityTerminal>

```

## F.205 XML\_ReaderAcrossHost

ReaderAcrossHost message in XML format

```

<ReaderAcrossHost version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <AcrossHostList size="8">
        <AcrossHostAction>
            <readerNo><!--req, xs: integer, card reader No., which is between 1 and
8--></readerNo>
            <submarineBackEnabled>
                <!--req, xs: boolean, whether to enable the cross-controller anti-
passing back function of the card reader-->
            </submarineBackEnabled>
        </AcrossHostAction>
    </AcrossHostList>
</ReaderAcrossHost>

```

## F.206 XML\_ResponseStatus

XML message about response status

```
<?xml version="1.0" encoding="utf-8"?>
<ResponseStatus version="2.0" xmlns="http://www.std-cgi.org/ver20/XMLSchema">
    <requestURL>
        <!--required, read-only, xs:string, request URL-->
    </requestURL>
    <statusCode>
        <!--required, read-only, xs:integer, status code: 0,1-OK, 2-Device Busy, 3-Device Error, 4-Invalid Operation, 5-Invalid XML Format, 6-Invalid XML Content, 7-Reboot Required, 9-Additional Error-->
    </statusCode>
    <statusString>
        <!--required, read-only, xs:string, status description: OK, Device Busy, Device Error, Invalid Operation, Invalid XML Format, Invalid XML Content, Reboot, Additional Error-->
    </statusString>
    <subStatusCode>
        <!--required, read-only, xs:string, describe the error reason in detail-->
    </subStatusCode>
    <MErrCode>
        <!--optional, xs:string, error code categorized by functional modules, e.g., 0x12345678-->
    </MErrCode>
    <MErrDevSelfEx>
        <!--optional, xs:string, extension field of MErrCode. It is used to define the custom error code, which is categorized by functional modules-->
    </MErrDevSelfEx>
</ResponseStatus>
```

## F.207 XML\_ServerDevice

ServerDevice message in XML format

```
<ServerDevice version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <ipAddr><!--req, xs: string, IP address of the cross-controller anti-passing back server--></ipAddr>
    <port><!--req, xs: string, port No. of the cross-controller anti-passing back server--></port>
</ServerDevice>
```

## F.208 XML\_StartReaderInfo

StartReaderInfo message in XML format

```
<StartReaderInfo version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <hostNo><!--req, xs: integer, access controller No., min="1" max="64"--></hostNo>
        <readerNo><!--req, xs: integer, card reader No., min="1" max="8"--></readerNo>
</StartReaderInfo>
```

### F.209 XML\_SubmarineBack

SubmarineBack message in XML format

```
<SubmarineBack version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <enabled><!--req, xs: boolean, whether to specify this access controller as
the cross-controller anti-passing back server--></enabled>
</SubmarineBack>
```

### F.210 XML\_SubmarineBackHostInfo

SubmarineBackHostInfo message in XML format

```
<SubmarineBackHostInfo version="2.0" xmlns="http://www.isapi.org/ver20/
XMLSchema">
    <HostInfoList size="16">
        <Action>
            <deviceNo><!--req, xs: integer, device No., which is between 1 and 64--></deviceNo>
                <serial><!--req, xs: string, device serial No., min="9" max="9"--></serial>
            </Action>
        </HostInfoList>
    </SubmarineBackHostInfo>
```

### F.211 XML\_SubmarineBackMode

SubmarineBackMode message in XML format

```
<SubmarineBackMode version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <mode>
        <!--req, xs:string, anti-passing back mode: "disable"-anti-passing back is
disabled, "internetCommunicate"-based on network, "cardReadAndWrite"-based on
card-->
    </mode>
    <rule>
        <!--req, xs:string, anti-passing back rule: "line"-route anti-passing back,
"inOrOut"-entrance/exit anti-passing back. This node is invalid when the mode
is set to "disable"-->
    </rule>
```

```
<sectionID>
    <!--req, xs:integer, section ID, which is between 1 and 100. This node is
valid when mode is "cardReadAndWrite", and only one section ID can be
configured for one configuration-->
</sectionID>
</SubmarineBackMode>
```

### F.212 XML\_SubmarineBackReader

SubmarineBackReader message in XML format

```
<SubmarineBackReader version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
    <selfHostNo><!--req, xs:integer, access controller No. of the configuration
object, which is between 1 and 64--></selfHostNo>
    <selfReaderNo><!--req, xs:integer, card reader No. of the configuration
object, which is between 1 and 8--></selfReaderNo>
    <FollowReaderList size="16">
        <Action>
            <followHostNo><!--req, xs:integer, following access controller No., which
is between 1 and 64--></followHostNo>
            <followReaderNo><!--req, xs:integer, following card reader No., which is
between 1 and 8--></followReaderNo>
        </Action>
    </FollowReaderList>
</SubmarineBackReader>
```

### F.213 XML\_SubscribeEvent

SubscribeEvent message in XML format

```
<SubscribeEvent version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema" >
    <heartbeat>
        <!--optional, xs:integer, heartbeat interval, unit: second, the default
value is 30s-->
    </heartbeat>
    <eventMode>
        <!--required, xs:string, "all"-upload all alarms/events, "list"-upload
specified alarm/event-->
    </eventMode>
    <EventList>
        <Event><!--uploading mode of specified alarm/event, this node exists only
when eventMode is "list"-->
            <type>
                <!--required, xs:string, alarm/event types, which are obtained from the
capability, refer to Alarm/Event Types for Subscription for its values-->
            </type>
            <minorAlarm>
                <!--opt, xs:string, minor alarm type: "0x400,0x401,0x402,0x403", see
details in Access Control Event Type. This node is required when type is
-->
            </minorAlarm>
        </Event>
    </EventList>
</SubscribeEvent>
```

```
"AccessControllerEvent"-->
    </minorAlarm>
    <minorException>
        <!--opt, xs:string, minor exception type: "0x400,0x401,0x402,0x403",
see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
    </minorException>
    <minorOperation>
        <!--opt, xs:string, minor operation type: "0x400,0x401,0x402,0x403",
see details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
    </minorOperation>
    <minorEvent>
        <!--opt, xs:string, minor event type: "0x01,0x02,0x03,0x04", see
details in Access Control Event Type. This node is required when type is
"AccessControllerEvent"-->
    </minorEvent>
    <pictureURLType>
        <!--opt, xs:string, alarm picture format: "binary"-binary, "localURL"-device
local URL, "cloudStorageURL"-cloud storage URL-->
    </pictureURLType>
</Event>
</EventList>
<channels>
    <!--optional, xs:string, event linked channel information, and multiple
channels can be linked, each channel is separated by comma, e.g., "1,2,3,4..."-->
</channels>
<channels>
    <!--optional, xs:string, specify channels (each channel is separated by
comma, e.g., "1,2,3,4...") to be armed, this node does not exist if you want to
arm all channels, and if this node exists, the sub node <channels> in the node
<Event> is invalid-->
</channels>
<identityKey max="64"/>
<!--opt, xs: string, interaction command of subscription, supports
subscribing comparison results of face picture library (importing with this
command), the maximum length is 64-->
</SubscribeEvent>
```

## Appendix G. Response Codes of Text Protocol

The response codes returned during the text protocol integration is based on the status codes of HTTP. 7 kinds of status codes are predefined, including 1 (OK), 2 (Device Busy), 3 (Device Error), 4 (Invalid Operation), 5 (Invalid Message Format), 6 (Invalid Message Content), and 7 (Reboot Required). Each kind of status code contains multiple sub status codes, and the response codes are in a one-to-one correspondence with the sub status codes.

### **StatusCode=1**

SubStatusCode	Error Code	Description
ok	0x1	Operation completed.
riskPassword	0x10000002	Risky password.
armProcess	0x10000005	Arming process.

### **StatusCode=2**

Sub Status Code	Error Code	Description
noMemory	0x20000001	Insufficient memory.
serviceUnavailable	0x20000002	The service is not available.
upgrading	0x20000003	Upgrading.
deviceBusy	0x20000004	The device is busy or no response.
reConnectIpc	0x20000005	The video server is reconnected.
transferUpgradePackageFailed	0x20000006	Transmitting device upgrade data failed.
startUpgradeFailed	0x20000007	Starting upgrading device failed.
getUpgradeProcessfailed.	0x20000008	Getting upgrade status failed.
certificateExist	0x2000000B	The Authentication certificate already exists.

**StatusCode=3**

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
deviceError	0x30000001	Hardware error.
badFlash	0x30000002	Flash operation error.
28181Uninitialized	0x30000003	The 28181 configuration is not initialized.
socketConnectError	0x30000005	Connecting to socket failed.
receiveError	0x30000007	Receive response message failed.
deletePictureError	0x3000000A	Deleting picture failed.
pictureSizeExceedLimit	0x3000000C	Too large picture size.
clearCacheError	0x3000000D	Clearing cache failed.
updateDatabaseError	0x3000000F	Updating database failed.
searchDatabaseError	0x30000010	Searching in the database failed.
writeDatabaseError	0x30000011	Writing to database failed.
deleteDatabaseError	0x30000012	Deleting database element failed.
searchDatabaseElementError	0x30000013	Getting number of database elements failed.
cloudAutoUpgradeException	0x30000016	Downloading upgrade packet from cloud and upgrading failed.
HBPException	0x30001000	HBP exception.
UDEPException	0x30001001	UDEP exception
elasticSearchException	0x30001002	Elastic exception.
kafkaException	0x30001003	Kafka exception.
HBaseException	0x30001004	Hbase exception.
sparkException	0x30001005	Spark exception.
yarnException	0x30001006	Yarn exception.
cacheException	0x30001007	Cache exception.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
trafficException	0x30001008	Monitoring point big data server exception.
faceException	0x30001009	Human face big data server exception.
SSDFileSystemIsError	0x30001013	SSD file system error (Error occurs when it is non-Ext4 file system)
insufficientSSDCapacityForFPD	0x30001014	Insufficient SSD space for person frequency detection.
wifiException	0x3000100A	Wi-Fi big data server exception
structException	0x3000100D	Video parameters structure server exception.
noLinkageResource	0x30001015	Insufficient linkage resources.
engineAbnormal	0x30002015	Engine exception.
engineInitialization	0x30002016	Initializing the engine.
algorithmLoadingFailed	0x30002017	Loading the model failed.
algorithmDownloadFailed	0x30002018	Downloading the model failed.
algorithmDecryptionFailed	0x30002019	Decrypting the model failed.
unboundChannel	0x30002020	Delete the linked channel to load the new model.
unsupportedResolution	0x30002021	Invalid resolution.
unsupportedSteamType	0x30002022	Invalid stream type.
insufficientDecRes	0x30002023	Insufficient decoding resources.
insufficientEnginePerformance	0x30002024	Insufficient engine performance (The number of channels to be analyzed exceeds the engine's capability).
improperResolution	0x30002025	Improper resolution (The maximum resolution allowed is 4096×4096).

Sub Status Code	Error Code	Description
improperPicSize	0x30002026	Improper picture size (The maximum size allowed is 5MB).
URLDownloadFailed	0x30002027	Downloading the picture via the URI failed.
unsupportedImageFormat	0x30002028	Invalid picture format (Only JPG is supported currently).
unsupportedPollingIntervalTime	0x30002029	Invalid polling interval (The interval should be more than 10s).
exceedImagesNumber	0x30002030	The number of pictures exceeds the limit (The platform can apply 1 to 100 picture URIs per time, the maximum number allowed is 100).
unsupportedMPID	0x30002031	The applied MPID does not exist in the device, so updating this MPID is not supported.
modelPackageNotMatchLabel	0x30002032	The model and the description file mismatch.
modelPackageNotMatchTask	0x30002033	The task and the model type mismatch.
insufficientSpace	0x30002034	Insufficient space (When the number of model packages does not reach the maximum number allowed but their size together exceeds the free space, the model packages cannot be added).
engineUnLoadingModelPackage	0x30002035	Applying the task failed. This engine is not linked to a model package (Canceling the linkage failed, this engine is not linked to a model package).
engineWithModelPackage	0x30002036	Linking the engine to this model package failed. The engine has been linked to

Sub Status Code	Error Code	Description
		another model package. Please cancel their linkage first.
modelPackageDelete	0x30002037	Linking the model package failed. The model package has been deleted.
deleteTaskFailed	0x30002038	Deleting the task failed (It is returned when the user fails to end a task).
modelPackageNumberslimited	0x30002039	Adding the model package failed. The number of model package has reached the maximum number allowed.
modelPackageDeleteFailed	0x30002040	Deleting the model package failed.
noArmingResource	0x30001016	Insufficient arming resources.
calibrationTimeout	0x30002051	Calibration timed out.
captureTimeout	0x30006000	Data collection timed out.
lowScore	0x30006001	Low quality of collected data.
uploadingFailed	0x30007004	Uploading failed.

**StatusCode=4**

Sub Status Code	Error Code	Description
notSupport	0x40000001	Not supported.
lowPrivilege	0x40000002	No permission.
badAuthorization	0x40000003	Authentication failed.
methodNotAllowed	0x40000004	Invalid HTTP method.
notSetHdiskRedund	0x40000005	Setting spare HDD failed.
invalidOperation	0x40000006	Invalid operation.
notActivated	0x40000007	Inactivated.
hasActivated	0x40000008	Activated.
certificateAlreadyExist	0x40000009	The certificate already exists.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
operateFailed	0x4000000F	Operation failed.
USBNotExist	0x40000010	USB device is not connected.
upgradePackageMorethan2GB	0x40001000	Up to 2GB upgrade package is allowed to be uploaded.
IDNotExist	0x40001001	The ID does not exist.
interfaceOperationError	0x40001002	API operation failed.
synchronizationError	0x40001003	Synchronization failed.
synchronizing	0x40001004	Synchronizing.
importError	0x40001005	Importing failed.
importing	0x40001006	Importing.
fileAlreadyExists	0x40001007	The file already exists.
invalidID	0x40001008	Invalid ID.
backupnodeNotAllowed	0x40001009	Accessing to backup node is not allowed.
exportingError	0x4000100A	Exporting failed.
exporting	0x4000100B	Exporting.
exportEnded	0x4000100C	Exporting stopped.
exported	0x4000100D	Exported.
IPOccupied	0x4000100E	The IP address is already occupied.
IDAlreadyExists	0x4000100F	The ID already exists.
exportItemsExceedLimit	0x40001010	No more items can be exported.
noFiles	0x40001011	The file does not exist.
beingExportedByAnotherUser	0x40001012	Being exported by others.
needReAuthentication	0x40001013	Authentication is needed after upgrade.
unitAddNotOnline	0x40001015	The added data analysis server is offline.
unitControl	0x40001016	The data analysis server is already added.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
analysis unitFull	0x40001017	No more data analysis server can be added.
unitIDError	0x40001018	The data analysis server ID does not exist.
unitExit	0x40001019	The data analysis server already exists in the list.
unitSearch	0x4000101A	Searching data analysis server in the list failed.
unitNotOnline	0x4000101B	The data analysis server is offline.
unitInfoError	0x4000101C	Getting data analysis server information failed.
unitGetNodeInfoError	0x4000101D	Getting node information failed.
unitGetNetworkInfoError	0x4000101E	Getting the network information of data analysis server failed
unitSetNetworkInfoError	0x4000101F	Setting the network information of data analysis server failed
setSmartNodeInfoError	0x40001020	Setting node information failed.
setUnitNetworkInfoError	0x40001021	Setting data analysis server network information failed.
unitRestartCloseError	0x40001022	Rebooting or shutting down data analysis server failed.
virtualIPnotAllowed	0x40001023	Adding virtual IP address is not allowed.
unitInstalled	0x40001024	The data analysis server is already installed.
badSubnetMask	0x40001025	Invalid subnet mask.
uintVersionMismatched	0x40001026	Data analysis server version mismatches.
deviceModelMismatched	0x40001027	Adding failed. Device model mismatches.
unitAddNotSelf	0x40001028	Adding peripherals is not allowed.
noValidUnit	0x40001029	No valid data analysis server.
unitNameDuplicate	0x4000102A	Duplicated data analysis server name.
deleteUnitFirst	0x4000102B	Delete the added data analysis server of the node first.
getLocalInfoFailed	0x4000102C	Getting the server information failed.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
getClientAddedNodeFailed	0x4000102D	Getting the added node information of data analysis server failed.
taskExit	0x4000102E	The task already exists.
taskInitError	0x4000102F	Initializing task failed.
taskSubmitError	0x40001030	Submitting task failed.
taskDelError	0x40001031	Deleting task failed.
taskPauseError	0x40001032	Pausing task failed.
taskContinueError	0x40001033	Starting task failed.
taskSeverNoCfg	0x40001035	Full-text search server is not configured.
taskPicSeverNoCfg	0x40001036	The picture server is not configured.
taskStreamError	0x40001037	Streaming information exception.
taskRecSDK	0x40001038	History recording is not supported.
taskCasaError	0x4000103A	Cascading is not supported.
taskVCARuleError	0x4000103B	Invalid VCA rule.
taskNoRun	0x4000103C	The task is not executed.
unitLinksNoStorageNode	0x4000103D	No node is linked with the data analysis server. Configure the node first.
searchFailed	0x4000103E	Searching video files failed.
searchNull	0x4000103F	No video clip.
userScheOffline	0x40001040	The task scheduler service is offline.
updateTypeUnmatched	0x40001041	The upgrade package type mismatches.
userExist	0x40001043	The user already exists.
userCannotDelAdmin	0x40001044	The administrator cannot be deleted.
userInexistence	0x40001045	The user name does not exist.
userCannotCreateAdmin	0x40001046	The administrator cannot be created.
monitorCamExceed	0x40001048	Up to 3000 cameras can be added.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
monitorCunitOverLimit	0x40001049	Adding failed. Up to 5 lower-levels are supported by the control center.
monitorReginOverLimit	0x4000104A	Adding failed. Up to 5 lower-levels are supported by the area.
monitorArming	0x4000104B	The camera is already armed. Disarm the camera and try again.
monitorSyncCfgNotSet	0x4000104C	The system parameters are not configured.
monitorFdSyncing	0x4000104E	Synchronizing. Try again after completing the synchronization.
monitorParseFailed	0x4000104F	Parsing camera information failed.
monitorCreatRootFailed	0x40001050	Creating resource node failed.
deleteArmingInfo	0x40001051	The camera is already . Disarm the camera and try again.
cannotModify	0x40001052	Editing is not allowed. Select again.
cannotDel	0x40001053	Deletion is not allowed. Select again.
deviceExist	0x40001054	The device already exists.
IPErrorConnectFailed	0x40001056	Connection failed. Check the network port.
cannotAdd	0x40001057	Only the capture cameras can be added.
serverExist	0x40001058	The server already exists.
fullTextParamError	0x40001059	Incorrect full-text search parameters.
storParamError	0x4000105A	Incorrect storage server parameters.
picServerFull	0x4000105B	The storage space of picture storage server is full.
NTPUnconnect	0x4000105C	Connecting to NTP server failed. Check the parameters.
storSerConnectFailed	0x4000105D	Connecting to storage server failed. Check the network port.
storSerLoginFailed	0x4000105E	Logging in to storage server failed. Check the user name and password.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
searchSerConnectFailed	0x4000105F	Connecting to full-text search server failed. Check the network port.
searchSerLoginFailed	0x40001060	Logging in to full-text search server failed. Check the user name and password.
kafkaConnectFailed	0x40001061	Connecting to Kafka failed. Check the network port.
mgmtConnectFailed	0x40001062	Connecting to system failed. Check the network port.
mgmtLoginFailed	0x40001063	Logging in to system failed. Check the user name and password.
TDAConnectFailed	0x40001064	Connecting to traffic data access server failed. Checking the server status.
86sdkConnectFailed	0x40001065	Connecting to listening port of iVMS-8600 System failed. Check the parameters.
nameExist	0x40001066	Duplicated server name.
batchProcessFailed	0x40001067	Processing in batch failed.
IDNotExist	0x40001068	The server ID does not exist.
serviceNumberReachesLimit	0x40001069	No more service can be added.
invalidServiceType.	0x4000106A	Invalid service type.
clusterGetInfo	0x4000106B	Getting cluster group information failed.
clusterDelNode	0x4000106C	Deletion node failed.
clusterAddNode	0x4000106D	Adding node failed.
clusterInstalling	0x4000106E	Creating cluster...Do not operate.
clusterUninstall	0x4000106F	Reseting cluster...Do not operate.
clusterInstall	0x40001070	Creating cluster failed.
clusterIpError	0x40001071	Invalid IP address of task scheduler server.
clusterNotSameSeg	0x40001072	The main node and sub node must be in the same network segment.
clusterVirIpError	0x40001073	Automatically getting virtual IP address failed. Enter manually.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
clusterNodeUnadd	0x40001074	The specified main (sub) node is not added.
clusterNodeOffline	0x40001075	The task scheduler server is offline.
nodeNotCurrentIP	0x40001076	The analysis node of the current IP address is required when adding main and sub nodes.
addNodeNetFailed	0x40001077	Adding node failed. The network disconnected.
needTwoMgmtNode	0x40001078	Two management nodes are required when adding main and sub nodes.
ipConflict	0x40001079	The virtual IP address and data analysis server's IP address conflicted.
ipUsed	0x4000107A	The virtual IP address has been occupied.
cloudAlalyseOnline	0x4000107B	The cloud analytic server is online.
virIP&mainIPnotSameNetSegment	0x4000107C	The virtual IP address is not in the same network segment with the IP address of main/ sub node.
getNodeDispatchInfoFailed	0x4000107D	Getting node scheduler information failed.
unableModifyManagementNetworkIP	0x4000107E	Editing management network interface failed. The analysis board is in the cluster.
notSpecifyVirtualIP	0x4000107F	Virtual IP address should be specified for main and sub cluster.
armingFull	0x40001080	No more device can be armed.
armingNoFind	0x40001081	The arming information does not exist.
disArming	0x40001082	Disarming failed.
getArmingError	0x40001084	Getting arming information failed.
refreshArmingError	0x40001085	Refreshing arming information failed.
ArmingPlateSame	0x40001086	The license plate number is repeatedly armed.
ArmingParseXLSError	0x40001087	Parsing arming information file failed.
ArmingTimeError	0x40001088	Invalid arming time period.
ArmingSearchTimeError	0x40001089	Invalid search time period.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
armingRelationshipReachesLimit	0x4000108A	No more relation can be created.
duplicateArmingName	0x4000108B	The relation name already exists.
noMoreArmingListAdded	0x4000108C	No more blocklist library can be armed.
noMoreCamerasAdded	0x4000108D	No more camera can be armed.
noMoreArmingListAddedWithCamera	0x4000108E	No more library can be linked to the camera.
noMoreArmingPeriodAdded	0x4000108F	No more time period can be added to the arming schedule.
armingPeriodsOverlapped	0x40001090	The time periods in the arming schedule are overlapped.
noArmingAlarmInfo	0x40001091	The alarm information does not exist.
armingAlarmUnRead	0x40001092	Getting number of unread alarms failed.
getArmingAlarmError	0x40001093	Getting alarm information failed.
searchByPictureTimedOut	0x40001094	Searching picture by picture timeout. Search again.
comparisonTimeRangeError	0x40001095	Comparison time period error.
selectMonitorNumberUpperLimit	0x40001096	No more monitoring point ID can be filtered.
noMoreComparisonTasksAdded	0x40001097	No more comparison task can be executed at the same time.
GetComparisonResultFailed	0x40001098	Getting comparison result failed.
comparisonTypeError	0x40001099	Comparison type error.
comparisonUnfinished	0x4000109A	The comparison is not completed.
facePictureModelInvalid	0x4000109B	Invalid face model.
duplicateLibraryName.	0x4000109C	The library name already exists.
noRecord	0x4000109D	No record found.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
countingRecordsFailed.	0x4000109E	Calculate the number of records failed.
getHumanFaceFrameFailed	0x4000109F	Getting face thumbnail from the picture failed.
modelingFailed.	0x400010A0	Modeling face according to picture URL failed.
1V1FacePictureComparisonFailed	0x400010A1	Comparison 1 VS 1 face picture failed.
libraryArmed	0x400010A2	The blocklist library is armed.
licenseExceedLimit	0x400010A3	Dongle limited.
licenseExpired	0x400010A4	Dongle expired.
licenseDisabled	0x400010A5	Unavailable dongle.
licenseNotExist	0x400010A6	The dongle does not exist.
SessionExpired	0x400010A7	Session expired .
beyondConcurrentLimit	0x400010A8	Out of concurrent limit.
stopSync	0x400010A9	Synchronization stopped.
getProgressFaild	0x400010AA	Getting progress failed.
uploadExtraCaps	0x400010AB	No more files can be uploaded.
timeRangeError	0x400010AC	Time period error.
dataPortNotConnected	0x400010AD	The data port is not connected.
addClusterNodeFailed	0x400010AE	Adding to the cluster failed. The device is already added to other cluster.
taskNotExist	0x400010AF	The task does not exist.
taskQueryFailed	0x400010B0	Searching task failed.
modifyTimeRuleFailed	0x400010B2	The task already exists. Editing time rule is not allowed.
modifySmartRuleFailed	0x400010B3	The task already exists. Editing VAC rule is not allowed.
queryHistoryVideoFailed	0x400010B4	Searching history video failed.
addDeviceFailed	0x400010B5	Adding device failed.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
addVideoFailed	0x400010B6	Adding video files failed.
deleteAllVideoFailed	0x400010B7	Deleting all video files failed.
createVideoIndexFailed	0x400010B8	Indexing video files failed.
videoCheckTypeFailed	0x400010B9	Verifying video files types failed.
configStructuredAddressesFailed	0x400010BA	Configuring IP address of structured server failed.
configPictureServerAddressFailed	0x400010BB	Configuring IP address of picture storaged server failed.
storageServiceIPNotExist	0x400010BD	The storage server IP address does not exist.
syncBackupDatabaseFailed	0x400010BE	Synchronizing sub database failed. Try again.
syncBackupNTPTimeFailed	0x400010BF	Synchronizing NTP time of sub server failed.
clusterNotSelectLoopbackAddress	0x400010C0	Loopbacl address is not supported by the main or sub cluster.
addFaceRecordFailed	0x400010C1	Adding face record failed.
deleteFaceRecordFailed	0x400010C2	Deleting face record failed.
modifyFaceRecordFailed	0x400010C3	Editing face record failed.
queryFaceRecordFailed	0x400010C4	Searching face record failed.
faceDetectFailed	0x400010C5	Detecting face failed.
libraryNotExist	0x400010C6	The library does not exist.
blackListQueryExporting	0x400010C7	Exporting matched blocklists.
blackListQueryExported	0x400010C8	The matched blocklists are exported.
blackListQueryStopExporting	0x400010C9	Exporting matched blocklists is stopped.
blackListAlarmQueryExporting	0x400010CA	Exporting matched blocklist alarms.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
blackListAlarmQueryExported	0x400010CB	The matched blocklists alarms are exported.
blackListAlarmQueryStopExporting	0x400010CC	Exporting matched blocklist alarms is stopped.
getBigDataCloudAnalysisFailed	0x400010CD	Getting big data cloud analytic information failed.
setBigDataCloudAnalysisFailed	0x400010CE	Configuring big data cloud analytic failed.
submitMapSearchFailed	0x400010CF	Submitting search by picture task failed.
controlRelationshipNotExist	0x400010D0	The relation does not exist.
getHistoryAlarmInfoFailed	0x400010D1	Getting history alarm information failed.
getFlowReportFailed	0x400010D2	Getting people counting report failed.
addGuardFailed	0x400010D3	Adding arming configuration failed.
deleteGuardFailed	0x400010D4	Deleting arming configuration failed.
modifyGuardFailed	0x400010D5	Editing arming configuration failed.
queryGuardFailed	0x400010D6	Searching arming configurations failed.
uploadUserSuperCaps	0x400010D7	No more user information can be uploaded.
bigDataServerConnectFailed	0x400010D8	Connecting to big data server failed.
microVideoCloudRequestInfoBuildFailed	0x400010D9	Adding response information of micro video cloud failed.
microVideoCloudResponseInfoBuildFailed	0x400010DA	Parsing response information of micro video cloud failed.
transcodingServerRequestInfoBuildFailed	0x400010DB	Adding response information of transcoding server failed.
transcodingServerResponseInfoParseFailed	0x400010DC	Parsing response information of transcoding server failed.
transcodingServerOffline	0x400010DD	Transcoding server is offline.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
microVideoCloudOffline	0x400010DE	Micro video cloud is offline.
UPSServerOffline	0x400010DF	UPS monitor server is offline.
statisticReportRequestInfoBuildFailed	0x400010E0	Adding response information of statistics report failed.
statisticReportResponseInfoParseFailed	0x400010E1	Parsing response information of statistics report failed.
DisplayConfigInfoBuildFailed	0x400010E2	Adding display configuration information failed.
DisplayConfigInfoParseFailed	0x400010E3	Parsing display configuration information failed.
DisplayConfigInfoSaveFailed	0x400010E4	Saving display configuration information failed.
notSupportDisplayConfigType	0x400010E5	The display configuration type is not supported.
passError	0x400010E7	Incorrect password.
upgradePackageLarge	0x400010EB	Too large upgrade package.
sessionUserReachesLimit	0x400010EC	No more user can log in via session.
ISO8601TimeFormatError	0x400010ED	Invalid ISO8601 time format.
clusterDissolutionFailed	0x400010EE	Deleting cluster failed.
getServiceNodeInfoFailed	0x400010EF	Getting service node information failed.
getUPSInfoFailed	0x400010F0	Getting UPS configuration information failed.
getDataStatisticsReportFailed	0x400010F1	Getting data statistic report failed.
getDisplayConfigInfoFailed	0x400010F2	Getting display configuration failed.
namingAnalysisBoardNotAllowed	0x400010F3	Renaming analysis board is not allowed.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
onlyDrawRegionsOfConvexPolygon	0x400010F4	Only drawing convex polygon area is supported.
bigDataServerResponseInfoParseFailed	0x400010F5	Parsing response message of big data service failed.
bigDataServerReturnFailed	0x400010F6	No response is returned by big data service.
microVideoReturnFailed	0x400010F7	No response is returned by micro video cloud service.
transcodingServerReturnFailed	0x400010F8	No response is returned by transcoding service.
UPSServerReturnFailed	0x400010F9	No response is returned by UPS monitoring service.
forwardingServerReturnFailed	0x400010FA	No response is returned by forwarding service.
storageServerReturnFailed	0x400010FB	No response is returned by storage service.
cloudAnalysisServerReturnFailed	0x400010FC	No response is returned by cloud analytic service.
modelEmpty	0x400010FD	No model is obtained.
mainAndBackupNodeCannotModifyManagementNetworkInterfaceIP	0x400010FE	Editing the management interface IP address of main node and backup node is not allowed.
IDTooLong	0x400010FF	The ID is too long.
pictureCheckFailed	0x40001100	Detecting picture failed.
pictureModelingFailed	0x40001101	Modeling picture failed.
setCloudAnalysisDefaultProvinceFailed	0x40001102	Setting default province of cloud analytic service failed.
InspectionAreasNumberExceedLimit	0x40001103	No more detection regions can be added.
picturePixelsTooLarge	0x40001105	The picture resolution is too high.
picturePixelsTooSmall	0x40001106	The picture resolution is too low.
storageServiceIPEmpty	0x40001107	The storage server IP address is required.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
bigDataServerRequestInfoBuildFail	0x40001108	Creating request message of big data service failed.
analysisTimedOut	0x40001109	Analysis time out.
high-performanceModeDisabled	0x4000110A	Please enable high-performance mode.
configuringUPSMonitorServerTimedOut	0x4000110B	Configuring the UPS monitoring server time out. Check IP address.
cloudAnalysisRequestInformationBuildFailed	0x4000110C	Creating request message of cloud analytic service failed.
cloudAnalysisResponseInformationParseFailed	0x4000110D	Parsing response message of cloud analytic service failed.
allCloudAnalysisInterfaceFailed	0x4000110E	Calling API for cloud analytic service failed.
cloudAnalysisModelCompareFailed	0x4000110F	Model comparison of cloud analytic service failed.
cloudAnalysisFacePictureQualityRatingFailed	0x40001110	Getting face quality grading of cloud analytic service failed.
cloudAnalysisExtractFeaturePointsFailed	0x40001111	Extracting feature of cloud analytic service failed.
cloudAnalysisExtractPropertyFailed	0x40001112	Extracting property of cloud analytic service failed.
getAddedNodeInformationFailed	0x40001113	Getting the added nodes information of data analysis server failed.
noMoreAnalysisUnitsAdded	0x40001114	No more data analysis servers can be added.
detectionAreaInvalid	0x40001115	Invalid detection region.
shieldAreaInvalid	0x40001116	Invalid shield region.
noMoreShieldAreasAdded	0x40001117	No more shield region can be drawn.
onlyAreaOfRectangleShapeAllowed	0x40001118	Only drawing rectangle is allowed in detection area.
numberReachedLimit	0x40001119	Number reached the limit.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
wait1~3MinutesGetIPAfterSetupDHCP	0x4000111A	Wait 1 to 3 minutes to get IP address after configuring DHCP.
plannedTimeMustbeHalfAnHour	0x4000111B	Schedule must be half an hour.
oneDeviceCannotBuildCluster	0x4000111C	Creating main and backup cluster requires at least two devices.
updatePackageFileNotUploaded	0x4000111E	Upgrade package is not uploaded.
highPerformanceTasksNotSupportDrawingDetectionRegions	0x4000111F	Drawing detection area is not allowed under high-performance mode.
controlCenterIDDoesNotExist	0x40001120	The control center ID does not exist.
regionIDDoesNotExist	0x40001121	The area ID does not exist.
licensePlateFormatError	0x40001122	Invalid license plate format.
managementNodeDoesNotSupportThisOperation	0x40001123	The operation is not supported.
searchByPictureResourceNotConfiged	0x40001124	The conditions for searching picture by picture are not configured.
videoFileEncapsulationFormatNotSupported	0x40001125	The video container format is not supported.
videoPackageFailure	0x40001126	Converting video container format failed.
videoCodingFormatNotSupported	0x40001127	Video coding format is not supported.
monitorOfDeviceArmingDeleteArmingInfo	0x40001129	The camera is armed. Disarm it and try again.
getVideoSourceTypeFailed	0x4000112A	Getting video source type failed.
smartRulesBuildFailed	0x4000112B	Creating VAC rule failed.
smartRulesParseFailed	0x4000112C	Parsing VAC rule failed.
timeRulesBuildFailed	0x4000112D	Creating time rule failed.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
timeRulesParseFailed	0x4000112E	Parsing time rule failed.
monitoInfoInvalid	0x4000112F	Invalid camera information.
addingFailedVersionMismatch	0x40001130	Adding failed. The device version mismatches.
theInformationReturnedAfterCloudAnalysisIsEmpty	0x40001131	No response is returned by the cloud analytic service.
selectingIpAddressOfHostAndSpareNodeFailedCheckTheStatus	0x40001132	Setting IP address for main node and backup node failed. Check the node status.
theSearchIdDoesNotExist	0x40001133	The search ID does not exist.
theSynchronizationIdDoesNotExist	0x40001134	The synchronization ID does not exist.
theUserIdDoesNotExist	0x40001136	The user ID does not exist.
theIndexCodeDoesNotExist	0x40001138	The index code does not exist.
theControlCenterIdDoesNotExist	0x40001139	The control center ID does not exist.
theAreaIdDoesNotExist	0x4000113A	The area ID does not exist.
theArmingLinkagIdDoesNotExist	0x4000113C	The arming relationship ID does not exist.
theListLibraryIdDoesNotExist	0x4000113D	The list library ID does not exist.
invalidCityCode	0x4000113E	Invalid city code.
synchronizingThePasswordOfSpareServerFailed	0x4000113F	Synchronizing backup system password failed.
editingStreamingTypeIsNotSupported	0x40001140	Editing streaming type is not supported.
switchingScheduledTaskToTemporaryTaskIsNotSupported	0x40001141	Switching scheduled task to temporary task is not supported.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
switchingTemporaryTaskToScheduledTaskIsNotSupported	0x40001142	Switching temporary task to scheduled task is not supported.
theTaskIsNotDispatchedOrItIsUpdating	0x40001143	The task is not dispatched or is updating.
thisTaskDoesNotExist	0x40001144	This task does not exist in the cloud analytic service.
duplicatedSchedule	0x40001145	Schedule period cannot be overlapped.
continuousScheduleWithSameAlgorithmTypeShouldBeMerged	0x40001146	The continuous schedule periods with same algorithm type should be merged.
invalidStreamingTimeRange	0x40001147	Invalid streaming time period.
invalidListLibraryType	0x40001148	Invalid list library type.
theNumberOfMatchedResultsShouldBeLargerThan0	0x40001149	The number of search results should be larger than 0.
invalidValueRangeOfSimilarity	0x4000114A	Invalid similarity range.
invalidSortingType	0x4000114B	Invalid sorting type.
noMoreListLibraryCanBeLinkedToTheDevice	0x4000114C	No more lists can be added to one device.
InvalidRecipientAddressFormat	0x4000114D	Invalid address format of result receiver.
creatingClusterFailedTheDongleIsNotPluggedIn	0x4000114E	Insert the dongle before creating cluster.
theURLIsTooLong	0x4000114F	No schedule configured for the task.
noScheduleIsConfiguredForTheTask	0x40001150	No schedule configured for the task.
theDongleIsExpired	0x40001151	Dongle has expired.
dongleException	0x40001152	Dongle exception.
invalidKey	0x40001153	Invalid authorization service key.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
decryptionFailed	0x40001154	Decrypting authorization service failed.
encryptionFailed	0x40001155	Encrypting authorization service failed.
AuthorizeServiceResponseError	0x40001156	Authorization service response exception.
incorrectParameter	0x40001157	Authorization service parameters error.
operationFailed	0x40001158	Operating authorization service error.
noAnalysisResourceOrNoDataInTheListLibrary	0x40001159	No cloud analytic resources or no data in the list library.
calculationException	0x4000115A	Calculation exception.
allocatingList	0x4000115B	Allocating list.
thisOperationIsNotSupportedByTheCloudAnalytics	0x4000115C	This operation is not supported by the cloud analytic serice.
theCloudAnalyticsIsInterrupted	0x4000115D	The operation of cloud analytic serice is interrupted.
theServiceIsNotReady	0x4000115E	The service is not ready.
searchingForExternalAPIFailed	0x4000115F	Searching external interfaces failed.
noOnlineNode	0x40001160	No node is online.
noNodeAllocated	0x40001161	No allocated node.
noMatchedList	0x40001162	No matched list.
allocatingFailedTooManyFacePictureLists	0x40001163	Allocation failed. Too many lists of big data service.
searchIsNotCompletedSearchAgain	0x40001164	Current searching is not completed. Search again.
allocatingListIsNotCompleted	0x40001165	Allocating list is not completed.
searchingForCloudAnalyticsResultsFailed	0x40001166	Searching cloud analytic serice overtime.
noDataOfTheCurrentLibraryFound	0x40001167	No data in the current library. Make sure there is data in the Hbase.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
noFacePictureLibraryIsArmed	0x40001168	No face picture library is armed for big data service.
noAvailableDataSlicingVersionInformationArmFirstAndSliceTheData	0x40001169	Invalid standard version information.
duplicatedOperationDataSlicingIsExecuting	0x4000116A	Slicing failed. Duplicated operation.
slicinDataFailedNoArmedFacePictureLibrary	0x4000116B	Slicing failed. No arming information in the face big data.
GenerateBenchmarkFileFailedSlicingAgain	0x4000116C	Generating sliced file failed. Slice again.
NonprimaryNodesProhibitedFromSlicingData	0x4000116D	Slicing is not allowed by the backup node.
NoReadyNodeToClusterServers	0x4000116E	Creating the cluster failed. No ready node.
NodeManagementServicesOffline	0x4000116F	The node management server is offline.
theCamera(s)OfTheControlCenterAreAlreadyArmed.DisarmThemFirst	0x40001170	Some cameras in control center are already armed. Disarm them and try again.
theCamera(s)OfTheAreaAreAlreadyArmed.DisarmThemFirst	0x40001171	Some cameras in this area are already armed. Disarm them and try again.
configuringHigh-frequencyPeopleDetectionFailed	0x40001172	Configuring high frequency people detection failed.
searchingForHigh-frequencyPeopleDetectionLogsFailed.	0x40001173	Searching detection event logs of high-frequency people detection failed.
gettingDetailsOfSearchedHigh-frequencyPeopleDetectionLogsFailed.	0x40001174	Getting the search result details of frequently appeared person alarms failed.

Sub Status Code	Error Code	Description
theArmedCamerasAlreadyExistInTheControlCenter	0x40001175	Some cameras in control center are already armed.
disarmingFailedTheCamerasNotArmed	0x40001177	Disarming failed. The camera is not armed.
noDataReturned	0x40001178	No response is returned by the big data service.
preallocFailure	0x40001179	Pre-allocating algorithm resource failed.
overDogLimit	0x4000117A	Configuration failed. No more resources can be pre-allocated.
analysisServicesDoNotSupport	0x4000117B	Not supported.
commandAndDispatchServiceError	0x4000117C	Scheduling service of cloud analytic serice error.
engineModuleError	0x4000117D	Engine module of cloud analytic serice error.
streamingServiceError	0x4000117E	Streaming component of cloud analytic serice error.
faceAnalysisModuleError	0x4000117F	Face analysis module of cloud analytic serice error.
vehicleAnalysisModuleError	0x40001180	Vehicle pictures analytic module of cloud analytic serice error.
videoStructuralAnalysisModuleError	0x40001181	Video structuring module of cloud analytic serice error.
postprocessingModuleError	0x40001182	Post-processing module of cloud analytic serice error.
frequentlyAppearedPersonAlarmIsAlreadyConfiguredForListLibrary	0x40001183	Frequently appeared person alarm is already armed for blocklist library.
creatingListLibraryFailed	0x40001184	Creating list library failed.
invalidIdentityKeyOfListLibrary	0x40001185	Invalid identity key of list library.
noMoreDevicesCanBeArmed	0x40001186	No more camera can be added.

Sub Status Code	Error Code	Description
settingAlgorithmTypeForDeviceFailed	0x40001187	Allocating task resource failed.
gettingHighFrequencyPersonDetectionAlarmInformationFailed	0x40001188	Setting frequently appeared person alarm failed.
invalidSearchConfition	0x40001189	Invalid result.
theTaskIsNotCompleted	0x4000118B	The task is not completed.
resourceOverRemainLimit	0x4000118C	No more resource can be pre-allocated.
frequentlyAppearedPersonAlarmsAlreadyConfiguredForTheCameraDisarmFirstAndTryAgain	0x4000118D	The frequently appeared person alarm of this camera is configured. Delete the arming information and try again.
switchtimedifflesslimit	0x4000123b	Time difference between power on and off should be less than 10 minutes.
associatedFaceLibNumOverLimit	0x40001279	Maximum number of linked face picture libraries reached.
noMorePeopleNumChangeRulesAdded	0x4000128A	Maximum number of people number changing rules reached.
noMoreViolentMotionRulesAdded	0x4000128D	Maximum number of violent motion rules reached.
noMoreLeavePositionRulesAdded	0x4000128E	Maximum number of leaving position rules reached.
SMRDiskNotSupportRa id	0x40001291	SMR disk does not support RAID.
OnlySupportHikAndCustomProtocol	0x400012A3	IPv6 camera can only be added via Device Network SDK or custom protocols.
vehicleEnginesNoResource	0x400012A6	Insufficient vehicle engine resources.
noMoreRunningRulesAdded	0x400012A9	Maximum number of running rules reached.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
noMoreGroupRulesAdded	0x400012AA	Maximum number of people gathering rules reached.
noMoreFailDownRulesAdded	0x400012AB	Maximum number of people falling down rules reached.
noMorePlayCellphoneRulesAdded	0x400012AC	Maximum number of playing cellphone rules reached.
ruleEventTypeDuplicate	0x400012C8	Event type duplicated.
noMoreRetentionRulesAdded	0x400015AD	Maximum number of people retention rules reached.
noMoreSleepOnDutyRulesAdded	0x400015AE	Maximum number of sleeping on duty rules reached.
polygonNotAllowCrossing	0x400015C2	Polygons are not allowed to cross.
configureRuleBeforeAdvanceParam	0x400015F8	Advanced parameters fail to be configured as no rule is configured, please configure rule information first.
behaviorCanNotPackToPic	0x40001603	The behavior model cannot be packaged as a picture algorithm.
noCluster	0x40001608	No cluster created.
NotAssociatedWithOwnChannel	0x400019C1	Current channel is not linked.
AITargetBPCaptureFail	0x400019C5	Capturing reference picture for AI target comparison failed.
AITargetBPToDSPFail	0x400019C6	Sending reference picture to DSP for AI target comparison failed.
AITargetBDuplicateName	0x400019C7	Duplicated name of reference picture for AI target comparison.
audioFileNameWrong	0x400019D0	Incorrect audio file name.
audioFileImportFail	0x400019D1	Importing audio file failed.
NonOperationalStandbyMachine	0x400019F0	Non-operational hot spare.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
MaximumNumberOfDevices	0x400019F1	The maximum number of devices reached.
StandbyMachineCannotBeDeleted	0x400019F2	The hot spare cannot be deleted.
alreadyRunning	0x40002026	The application program is running.
notRunning	0x40002027	The application program is stopped.
packetNotFound	0x40002028	The software packet does not exist.
alreadyExist	0x40002029	The application program already exists.
noMemory	0x4000202A	Insufficient memory.
invalidLicense	0x4000202B	Invalid License.
noClientCertificate	0x40002036	The client certificate is not installed.
noCACertificate	0x40002037	The CA certificate is not installed.
authenticationFailed	0x40002038	Authenticating certificate failed. Check the certificate.
clientCertificateExpired	0x40002039	The client certificate is expired.
clientCertificateRevocation	0x4000203A	The client certificate is revoked.
CACertificateExpired	0x4000203B	The CA certificate is expired.
CACertificateRevocation	0x4000203C	The CA certificate is revoked.
connectFail	0x4000203D	Connection failed.
loginNumExceedLimit	0x4000203F	No more user can log in.
HDMIResolutionIllegal	0x40002040	The HDMI video resolution cannot be larger than that of main and sub stream.
hdFormatFail	0x40002049	Formatting HDD failed.
formattingFailed	0x40002056	Formatting HDD failed.
encryptedFormattingFailed	0x40002057	Formatting encrypted HDD failed.
wrongPassword	0x40002058	Verifying password of SD card failed. Incorrect password.

Sub Status Code	Error Code	Description
audioIsPlayingPleaseWait	0x40002067	Audio is playing. Please wait.
twoWayAudioInProgressPleaseWait	0x40002068	Two-way audio in progress. Please wait.
calibrationPointNumFull	0x40002069	The maximum number of calibration points reached.
completeTheLevelCalibrationFirst	0x4000206A	The level calibration is not set.
completeTheRadarCameraCalibrationFirst	0x4000206B	The radar-camera calibration is not set.
pointsOnStraightLine	0x4000209C	Calibrating failed. The calibration points cannot be one the same line.
TValueLessThanOrEqualZero	0x4000209D	Calibration failed. The T value of the calibration points should be larger than 0.
HBDLibNumOverLimit	0x40002092	The number of human body picture libraries reaches the upper limit
theShieldRegionError	0x40002093	Saving failed. The shielded area should be the ground area where the shielded object is located.
theDetectionAreaError	0x40002094	Saving failed. The detection area should only cover the ground area.
invalidLaneLine	0x40002096	Saving failed. Invalid lane line.
enableITSFunctionOfThisChannelFirst	0x400020A2	Enable ITS function of this channel first.
noCloudStorageServer	0x400020C5	No cloud storage server
NotSupportWithVideoTask	0x400020F3	This function is not supported.
noDetectionArea	0x400050df	No detection area
armingFailed	0x40008000	Arming failed.
disarmingFailed	0x40008001	Disarming failed.
clearAlarmFailed	0x40008002	Clearing alarm failed.
bypassFailed	0x40008003	Bypass failed.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
bypassRecoverFailed	0x40008004	Bypass recovery failed.
outputsOpenFailed	0x40008005	Opening relay failed.
outputsCloseFailed	0x40008006	Closing relay failed.
registerTimeOut	0x40008007	Registering timed out.
registerFailed	0x40008008	Registering failed.
addedByOtherHost	0x40008009	The peripheral is already added by other security control panel.
alreadyAdded	0x4000800A	The peripheral is already added.
armedStatus	0x4000800B	The partition is armed.
bypassStatus	0x4000800C	Bypassed.
zoneNotSupport	0x4000800D	This operation is not supported by the zone.
zoneFault	0x4000800E	The zone is in fault status.
pwdConflict	0x4000800F	Password conflicted.
audioTestEntryFailed	0x40008010	Enabling audio test mode failed.
audioTestRecoveryFailed	0x40008011	Disabling audio test mode failed.
addCardMode	0x40008012	Adding card mode.
searchMode	0x40008013	Search mode.
addRemoterMode	0x40008014	Adding keyfob mode.
registerMode	0x40008015	Registration mode.
exDevNotExist	0x40008016	The peripheral does not exist.
theNumberOfExDevLimited	0x40008017	No peripheral can be added.
sirenConfigFailed	0x40008018	Setting siren failed.
chanCannotRepeatedBinded	0x40008019	This channel is already linked by the zone.
inProgramMode	0x4000801B	The keypad is in programming mode.
inPaceTest	0x4000801C	In pacing mode.
arming	0x4000801D	Arming.

Sub Status Code	Error Code	Description
masterSlavesEnable	0x4000802c	The main-sub relationship has taken effect, the sub radar does not support this operation.
forceTrackNotEnabled	0x4000802d	Mandatory tracking is disabled.
isNotSupportZoneConfigByLocalArea	0x4000802e	This area does not support the zone type.
alarmLineCross	0x4000802f	Trigger lines are overlapped.
zoneDrawingOutOfRange	0x40008030	The drawn zone is out of detection range.
alarmLineDrawingOutOfRange	0x40008031	The drawn alarm trigger line is out of detection range.
hasTargetInWarningArea	0x40008032	The warning zone already contains targets. Whether to enable mandatory arming?
radarModuleConnectFail	0x40008033	Radar module communication failed.
importCfgFilePasswordErr	0x40008034	Incorrect password for importing configuration files.
overAudioFileNumLimit	0x40008038	The number of audio files exceeds the limit.
audioFileNameIsLong	0x40008039	The audio file name is too long.
audioFormatIsWrong	0x4000803a	The audio file format is invalid.
audioFileIsLarge	0x4000803b	The size of the audio file exceeds the limit.
pircamCapTimeOut	0x4000803c	Capturing of pircam timed out.
pircamCapFail	0x4000803d	Capturing of pircam failed.
pircamIsCaping	0x4000803e	The pircam is capturing.
audioFileHasExisted	0x4000803f	The audio file already exists.
subscribeTypeErr	0x4000a016	This metadata type is not supported to be subscribed.
EISError	0x4000A01C	Electronic image stabilization failed. The smart event function is enabled.
jpegPicWithAppendDataError	0x4000A01D	Capturing the thermal graphic failed. Check if the temperature measurement parameters

Sub Status Code	Error Code	Description
		(emissivity, distance, reflective temperature) are configured correctly.
startAppFail	/	Starting running application program failed.
yuvconflict	/	The raw video stream conflicted.
overMaxAppNum	/	No more application program can be uploaded.
noFlash	/	Insufficient flash.
platMismatch	/	The platform mismatches.
emptyEventName	0x400015E0	Event name is empty.
sameEventName	0x400015E1	A same event name already exists.
emptyEventType	0x400015E2	Event type is required.
sameEventType	0x400015E3	A same event type already exists.
maxEventNameReached	0x400015E4	Maximum of events reached.
hotSpareNotAllowedExternalStorage	0x400015FC	External storage is not allowed when hot spare is enabled.
sameCustomProtocolName	0x400015FD	A same protocol name already exists.
maxPTZTriggerChannelReached	0x400015FE	Maximum of channels linked with PTZ reached.
POSCanotAddHolidayPlan	0x400015FF	No POS events during holidays.
eventTypeIsTooLong	0x40001600	Event type is too long.
eventNameIsTooLong	0x40001601	Event name is too long.
PerimeterEnginesNoResource	0x40001602	No more perimeter engines.
invalidProvinceCode	0x40001607	Invalid province code.

**Status Code=5**

Sub Status Code	Error Code	Description
badXmlFormat	0x50000001	Invalid XML format.

**StatusCode=6**

Sub Status Code	Error Code	Description
badParameters	0x60000001	Invalid parameter.
badHostAddress	0x60000002	Invalid host IP address.
badXmlContent	0x60000003	Invalid XML content.
badIPv4Address	0x60000004	Invalid IPv4 address.
badIPv6Address	0x60000005	Invalid IPv6 address.
conflictIPv4Address	0x60000006	IPv4 address conflicted.
conflictIPv6Address	0x60000007	IPv6 address conflicted.
badDomainName	0x60000008	Invalid domain name.
connectSreverFail	0x60000009	Connecting to server failed.
conflictDomainName	0x6000000A	Domain name conflicted.
badPort	0x6000000B	Port number conflicted.
portError	0x6000000C	Port error.
exportErrorData	0x6000000D	Importing data failed.
badNetMask	0x6000000E	Invalid sub-net mask.
badVersion	0x6000000F	Version mismatches.
badDevType	0x60000010	Device type mismatches.
badLanguage	0x60000011	Language mismatches.
incorrectUserNameOrPassword	0x60000012	Incorrect user name or password.
invalidStoragePoolOfCloudServer	0x60000013	Invalid storage pool. The storage pool is not configured or incorrect ID.
noFreeSpaceOfStoragePool	0x60000014	Storage pool is full.
riskPassword	0x60000015	Risky password.
UnSupportCapture	0x60000016	Capturing in 4096*2160 or 3072*2048 resolution is not supported when H.264+ is enabled.

Sub Status Code	Error Code	Description
userPwdLenUnder8	0x60000023	At least two kinds of characters, including digits, letters, and symbols, should be contained in the password.
userPwdNameSame	0x60000025	Duplicated password.
userPwdNameMirror	0x60000026	The password cannot be the reverse order of user name.
beyondARGSRangeLimit	0x60000027	The parameter value is out of limit.
DetectionLineOutofDetectionRegion	0x60000085	The rule line is out of region.
DetectionRegionError	0x60000086	Rule region error. Make sure the rule region is convex polygon.
DetectionRegionOutOfCountingRegion	0x60000087	The rule region must be marked as red frame.
PedalAreaError	0x60000088	The pedal area must be in the rule region.
DetectionAreaABError	0x60000089	The detection region A and B must be in the a rule frame.
ABRegionCannotIntersect	0x6000008a	Region A and B cannot be overlapped.
customHBPIDError	0x6000008b	Incorrect ID of custom human body picture library
customHBPIDRepeat	0x6000008c	Duplicated ID of custom human body picture library
dataVersionsInHBDLibMismatches	0x6000008d	Database versions mismatches of human body picture library
invalidHBPID	0x6000008e	Invalid human body picture PID
invalidHBDID	0x6000008f	Invalid ID of human body picture library
humanLibraryError	0x60000090	Error of human body picture library

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
humanLibraryNumError	0x60000091	No more human body picture library can be added
humanImagesNumError	0x60000092	No more human body picture can be added
noHumanInThePicture	0x60000093	Modeling failed, no human body in the picture
analysisEnginesNoResourceErr or	0x60001000	No analysis engine.
analysisEnginesUsageExced	0x60001001	The engine usage is overloaded.
PicAnalysisNoResourceError	0x60001002	No analysis engine provided for picture secondary recognition.
analysisEnginesLoadingError	0x60001003	Initializing analysis engine.
analysisEnginesAbnormaError	0x60001004	Analysis engine exception.
analysisEnginesFacelibImporting	0x60001005	Importing pictures to face picture library. Failed to edit analysis engine parameters.
analysisEnginesAssociatedChannel	0x60001006	The analysis engine is linked to channel.
smdEncodingNoResource	0x60001007	Insufficient motion detection encoding resources.
smdDecodingNoResource	0x60001008	Insufficient motion detection decoding resources.
diskError	0x60001009	HDD error.
diskFull	0x6000100a	HDD full.
facelibDataProcessing	0x6000100b	Handling face picture library data.
capturePackageFailed	0x6000100c	Capturing packet failed.
capturePackageProcessing	0x6000100d	Capturing packet.
noSupportWithPlaybackAbstract	0x6000100e	This function is not supported. Playback by video synopsis is enabled.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
insufficientNetworkBandwidth	0x6000100f	Insufficient network bandwidth.
tapeLibNeedStopArchive	0x60001010	Stop the filing operation of tape library first.
identityKeyError	0x60001011	Incorrect interaction command.
identityKeyMissing	0x60001012	The interaction command is lost.
noSupportWithPersonDensityDetect	0x60001013	This function is not supported. The people density detection is enabled.
ipcResolutionOverflow	0x60001014	The configured resolution of network camera is invalid.
ipcBitrateOverflow	0x60001015	The configured bit rate of network camera is invalid.
tooGreatTimeDifference	0x60001016	Too large time difference between device and server.
noSupportWithPlayback	0x60001017	This function is not supported. Playback is enabled.
channelNoSupportWithSMD	0x60001018	This function is not supported. Motion detection is enabled.
channelNoSupportWithFD	0x60001019	This function is not supported. Face capture is enabled.
illegalPhoneNumber	0x6000101a	Invalid phone number.
illegalCertificateNumber	0x6000101b	Invalid certificate No.
linkedCameraOutLimit	0x6000101c	Connecting camera timed out.
achieveMaxChannelLimit	0x6000101e	No more channels are allowed.
humanMisInfoFilterEnabledChanNumError	0x6000101f	No more channels are allowed to enable preventing false alarm.
humanEnginesNoResource	0x60001020	Insufficient human body analysis engine resources.
taskNumberOverflow	0x60001021	No more tasks can be added.

<b>Sub Status Code</b>	<b>Error Code</b>	<b>Description</b>
collisionTimeOverflow	0x60001022	No more comparison duration can be configured.
invalidTaskID	0x60001023	Invalid task ID.
eventNotSupport	0x60001024	Event subscription is not supported.
invalidEZVIZSecretKey	0x60001034	Invalid verification code for Hik-Connect.
needDoubleVerification	0x60001042	Double verification required
noDoubleVerificationUser	0x60001043	No double verification user
timeSpanNumOverLimit	0x60001044	Max. number of time buckets reached
channelNumOverLimit	0x60001045	Max. number of channels reached
noSearchIDResource	0x60001046	Insufficient searchID resources
noSupportDeleteStrangerLib	0x60001051	Deleting stranger library is not supported
noSupportCreateStrangerLib	0x60001052	Creating stranger library is not supported
behaviorAnalysisRuleInfoError	0x60001053	Behavior analysis rule parameters error.
safetyHelmetParamError	0x60001054	Hard hat parameters error.
OneChannelOnlyCanBindOneEngine	0x60001077	No more engines can be bound.
engineTypeMismatch	0x60001079	Engine type mismatched.
badUpgradePackage	0x6000107A	Invalid upgrade package.
AudioFileNameDuplicate	0x60001135	Duplicated audio file name.
CurrentAudioFileAIRuleInUseAIreadyDelete	0x60001136	The AI rule linkage related to current audio file has been deleted.
TransitionUseEmmc	0x60002000	Starting device failed. The EMMC is overused.

Sub Status Code	Error Code	Description
AdaptiveStreamNotEnabled	0x60002001	The stream self-adaptive function is not enabled.
AdaptiveStreamAndVariableBitRateEnabled	0x60002002	Stream self-adaptive and variable bitrate function cannot be enabled at the same time.
noSafetyHelmetRegion	0x60002023	The hard hat detection area is not configured (if users save their settings without configuring the arming area, they should be prompted to configure one).
unclosedSafetyHelmet	0x60002024	The hard hat detection is enabled (If users save their settings after deleting the arming area, they should be prompted to disable hard hat detection first and then delete the arming area).
width/heightRatioOfPictureError	0x6000202C	The width/height ratio of the uploaded picture should be in the range from 1:2 to 2:1.
PTZNotInitialized	0x6000202E	PTZ is not initialized.
PTZSelfChecking	0x6000202F	PTZ is self-checking.
PTZLocked	0x60002030	PTZ is locked.
advancedParametersError	0x60002031	Auto-switch interval in advanced parameters cannot be shorter than parking tolerance for illegal parking detection in speed dome rule settings.
resolutionError	0x60005003	Invalid resolution
deployExceedMax	0x60006018	The arming connections exceed the maximum number.
detectorTypeMismatch	0x60008000	The detector type mismatched.
nameExist	0x60008001	The name already exists.

Sub Status Code	Error Code	Description
uploadImageSizeError	0x60008016	The size of the uploaded picture is larger than 5 MB.
laneAndRegionOverlap	/	The lanes are overlapped.
unitConfigurationNotInEffect	/	Invalid unit parameter.
ruleAndShieldingMaskConflict	/	The line-rule region overlaps with the shielded area.
wholeRuleInShieldingMask	/	There are complete temperature measurement rules in the shielded area.
LogDiskNotSetReadOnlyInGroupMode	0x60001100	The log HDD in the HDD group cannot be set to read-only.
LogDiskNotSetReDundancyInGroupMode	0x60001101	The log HDD in the HDD group cannot be set to redundancy.
holidayNameContainChineseOrSpecialChar	0x60001080	No Chinese and special characters allowed in holiday name.
genderValueError	0x60001081	Invalid gender.
certificateTypeValueError	0x60001082	Invalid identification type.
personInfoExtendValueIsTooLong	0x60001083	The length of customized tags exceeds limit.
personInfoExtendValueContainsInvalidChar	0x60001084	Invalid characters are not allowed in customized tags of the face picture library.
excelHeaderError	0x60001085	Excel header error.
intelligentTrafficMutexWithHighFrames	0x60008014	Please disable all functions of traffic incident detection, violation enforcement, and traffic data collection, or adjust the video frame rate to that lower than 50 fps.
intelligentTrafficMutexWithHighFramesEx	0x60008018	Please disable all functions of traffic incident detection, violation enforcement, traffic data collection, and vehicle

Sub Status Code	Error Code	Description
		detection, or adjust the video frame rate to that lower than 50 fps.

### **StatusCode=7**

SubStatusCode	Error Code	Description
rebootRequired	0x70000001	Reboot to take effect.

## Appendix H. Error Codes Categorized by Functional Modules

The error codes returned during the text protocol integration is categorized by different functional modules. See the error codes, error descriptions, and debugging suggestions in the table below.

### Public Function Module (Error Codes Range: 0x00000000, from 0x00100001 to 0x001fffff)

Error String	Error Code	Description	Debugging Suggestion
success	0x00000000	Succeeded.	
deviceNotActivate d	0x00100001	The device is not activated.	Activate the device.
deviceNoPermission	0x00100002	Device operation failed. No permission.	Update user's permission.
deviceNotSupport	0x00100003	This function is not supported.	Check the device capability set and call the API corresponding to supported function.
deviceResourceN otEnough	0x00100004	Insufficient resources.	Release resources.
dataFormatError	0x00100005	Invalid message format.	
resetError	0x00100006	Restoring to factory settings failed. Reactivating device is required after the device is reboot as the Reset button may be stuck.	
parameterError	0x00100007	Incorrect parameter	
	0x00100100	Invalid channel	Check if the channel is valid.
	0x00100101	NPQ live view is not supported for stream encryption.	Replace streaming mode for stream encryption.
	0x00100102	No more channels are allowed for NPQ streaming.	Reduce NPQ streaming channels and try again.

Error String	Error Code	Description	Debugging Suggestion
	0x00100103	The stream type is not supported.	Check the requested stream type.
	0x00100104	The number of connections exceeded limit.	Reduce the number of streaming clients and try again.
	0x00100105	Not enough bandwidth.	Reduce the number of remote streaming channels.

### User Function Module (Error Codes Range: from 0x00200001 to 0x002fffff)

Error String	Error Code	Description	Debugging Suggestion
passwordError	0x00200001	Incorrect user name or password.	Check if the password is correct.
userNameNotExist	0x00200002	The account does not exist.	Check if the account exists, or add the account.
userNameLocked	0x00200003	The account is locked.	Wait for the device to unlock.
userNumLimited	0x00200004	The number of users allowed to log in exceeded the upper limit.	Log out.
lowPrivilege	0x00200005	No permissions for this operation	<p>For users operations, check the following situations:</p> <ul style="list-style-type: none"> <li>• Deleting your own account is not allowed.</li> <li>• Editing your own level or permission is not allowed.</li> <li>• Getting information about users with higher permission is not allowed.</li> <li>• Elevating the user's level or permission is not allowed.</li> </ul> <p>For other operations, check according to the following measures: If operations unrelated to user's permission configuration failed, you can check the user type and</p>

Error String	Error Code	Description	Debugging Suggestion
			permission, if not solved, contact the developers.
incorrectUserNameOrPassword	0x00200006	Incorrect user name or password	Check if the configured user name and password are matched. If not, contact the administrator to configure again. If the administrator forgets the password, reset the password of the device.
riskPassword	0x00200007	Risk password	Low password strength. Change password again.
passwordMustContainMorethan8Characters	0x00200008	The password length must be greater than or equal to 8.	Check if the password length is greater than or equal to 8. If not, change password again.
passwordLenNoMoreThan16	0x00200009	The password length cannot be greater than 16.	Check if the password length is greater than 16. If yes, change password again.
adminUserNotAllowedModify	0x0020000a	Editing admin information is not allowed.	Check if the edited account is admin.
confirmPasswordError	0x0020000b	Incorrect confirm password.	Check the confirm password.
passwordMustContainMorethan2Types	0x0020000c	The password must contain at least two or more of followings: numbers, lowercase, uppercase, and special characters.	Check if the configured password conforms the requirements.
passwordContainUserName	0x0020000d	The password cannot contain the user name.	Check if the password contains the user name.
userPwdNameMirror	0x0020000e	The password cannot be reversed user name.	Check if the password is reversed user name.

**Time Function Module (Error Codes Range: from 0x00300001 to 0x003fffff)**

Error String	Error Code	Description	Debugging Suggestion
manualAdjustmentFailed	0x00300001	Time synchronization failed.	
NTPError	0x00300002	Invalid NTP server address.	Check if the NTP server address is valid.
timeFormatError	0x00300003	Incorrect time format during time calibration.  For example, the time in ISO 8601 format should be "2018-02-01T19:54:04", but the applied time is "2018-02-01 19:54:04".	Incorrect message format or incorrect time format.
beyondTimeRangeLimit	0x00300004	The calibration time is not within the time range supported by the device.	Get the device capability and check if the configured time is within the time range supported by the device.
endtimeEarlierThanBeginTime	0x00300005	The start time of the validity period cannot be later than the end time.	Check if the start time and end time are valid.

**Network Function Module (Error Codes Range: from 0x00400001 to 0x004fffff)**

Error String	Error Code	Description	Debugging Suggestion
domainNameParseFailed	0x00400001	Parsing domain name failed.	
PPPOEConnectedFailed	0x00400002	Connecting PPPOE to the network failed.	
FTPConnectedFailed	0x00400003	The FTP server is disconnected.	
deviceIPConflicted	0x00400004	IP addresses of devices conflicted.	
libraryConnectedFailed	0x00400005	The image and video library is disconnected.	

Error String	Error Code	Description	Debugging Suggestion
fileUploadFailed	0x00400006	Uploading failed.	Check if the network connection is normal. If yes, contact after-sales.
storSerDownloadFileFailed	0x00400007	Downloading failed.	Check if the network connection is normal. If yes, contact after-sales.
storSerDownloadFileSizeZero	0x00400008	The size of file downloaded from the storage service is 0.	Check if the network connection is normal. If yes, contact after-sales.
storSerNotConfig	0x00400009	Storage service is not configured.	Check if the configuration is correct.
badHostAddress	0x0040000a	Host address error	Check if the configuration is correct.
badIPv4Address	0x0040000b	Incorrect IPv4 address.	Check if the configuration is correct.
badIPv6Address	0x0040000c	Incorrect IPv6 address.	Check if the configuration is correct.
conflictIPv4Address	0x0040000d	IPv4 address conflict.	Check the configuration status of IPV4 in the network.
conflictIPv6Address	0x0040000e	IPv6 address conflict	Check the configuration status of IPV6 in the network.
badDomainName	0x0040000f	Incorrect domain name.	Check if the configuration is correct.
connectSreverFail	0x00400010	Connecting to server failed.	Check if the network is normal and check if the configuration is correct.
conflictDomainName	0x00400011	Domain name conflict.	Check if the configuration is correct.
badPort	0x00400012	Port conflict.	Check if the configuration is correct.
portError	0x00400013	Port error	Check if the configuration is correct.

Error String	Error Code	Description	Debugging Suggestion
badNetMask	0x00400014	Subnet mask error	Check if the configuration is correct.
badVersion	0x00400015	Version mismatch	Check if the version is correct.
badDns	0x00400016	DNS error	Check if the configuration is correct.
badMTU	0x00400017	MTU error	Check if the configuration is correct.
badGateway	0x00400018	Wrong gateway	Check if the configuration is correct.
urlDownloadFail	0x00400019	Downloading via URL failed.	Check if the network is normal and check if the URL is correct.
deployExceedMax	0x0040001a	The number of armed channels exceeds the maximum number of connections.	Get the supported maximum number of arming and the number of armed channels.

### Maintenance Function Module (Error Codes Range: from 0x00500001 to 0x005ffff)

Error String	Error Code	Description	Debugging Suggestion
upgradeXMLForm atError	0x00500001	Incorrect XML upgrading request.	Check if the upgrade file is correct. If the file is correct, try the local upgrade.
upgradeContentEr ror	0x00500002	Incorrect upgrading request content.	Check if the upgrade file is correct. If the file is correct, try the local upgrade.
noUpgradePermis sion	0x00500003	No upgrade permission.	Switch to admin account or ask admin for advanced operation permission.
upgrading	0x00500004	Upgrading...	Wait for the upgrade to complete.
receiveUpgradePa ckageError	0x00500005	Receiving upgrade package failed.	Check if the network is normal.

Error String	Error Code	Description	Debugging Suggestion
upgradePackageLanguageMismatch	0x00500006	Upgrade package language mismatch.	Check the language type of upgrade package and the device.
upgradePackageMismatch	0x00500007	Upgrade file does not match with the device type.	Check the type of upgrade package and device.
OEMCodeMismatch	0x00500008	Upgrade package error. The OEM code mismatch.	Contact after-sales to get the correct upgrade package.
versionMismatch	0x00500009	Upgrade file version mismatch.	Contact after-sales to get the correct upgrade package.
upgradeHalfFailed	0x0050000c	Error occurred in the halfway of device upgrading. Flash error or cache error.	
deviceParameterImportFailed	0x0050000d	Importing device parameters failed. Device model, version, or platform mismatches.	
deviceEncryptionError	0x0050000e	Upgrade package mismatches. Device encryption error.	
SDCardFormatError	0x00500025	Formatting SD card failed.	
SDCardLoadFailed	0x00500026	Loading page failed after the SD card is inserted.	
NASFailed	0x00500027	Mounting NAS failed.	
hardDiskError	0x00500028	HDD exception (possible reasons: HDD does not exist, incompatible, encrypted, insufficient capacity, formatting exception, array exception, array incompatible, etc.)	
upgradeError	0x00500030	Upgrade error	

Error String	Error Code	Description	Debugging Suggestion
upgradePackageSizeMismath	0x00500032	Mismatch between the actual size of the downloaded upgrade package and the size in the upgrading request.	
upgradePackageSizeExceeded	0x00500033	The size of the package exceeded that of the partition.	
domainNameParseFailedForDownload	0x00500034	Parsing the domain name of the address for downloading failed.	
netWorkUnstable	0x00500035	Unstable network. Downloading timed out or the maximum number of attempts reached.	
digestValueMismatch	0x00500036	Mismatched digest value.	
signatureVerifyFailed	0x00500037	Verifying the signature failed.	
innerFormatError	0x00500038	Incorrect inner format of the upgrade package.	
memoryNotEnough	0x00500039	Insufficient memory.	
burnFailed	0x0050003a	Burning firmware failed.	
unknownError	0x0050003b	Unknown error occurred in the underlying APIs.	
userCancel	0x0050003c	User requested cancel of current operation.	
systemResume	0x0050003d	Upgrading failed. You can resume via the backup system or minimum system.	
	0x00500080	Upgrade file is not found.	Check if the upgrade package path is too long or if there is a correct upgrade

Error String	Error Code	Description	Debugging Suggestion
			package under the upgrade package path.
	0x00500081	Upgrade file does not match with the engine type.	Select the upgrade package matched with the device engine type.
	0x00500082	Parsing camera domain name failed.	Confirm if the device is correctly configured DNS service and if the camera domain is valid.
	0x00500083	Camera network is unreachable.	Confirm if the local network can access the network where the added channel located.

### Live View Module (Error Codes Range: from 0x00600001 to 0x006fffff)

Error String	Error Code	Description	Debugging Suggestion
liveViewFailed	0x00600001	Live view failed. The number of streaming channels exceeded limit.	
	0x00600002	Request packaging format exception.	Check the packaging format of requested live view.
	0x00600003	NPQ will be unavailable after enabling EHome 2.x.	When EHome 2.x is enable, use other live view mode.
	0x00600005	NPQ live view is not supported for channel-zero.	User other live view mode for channel-zero.
	0x00600007	Only virtual stream supports NPQ live view.	Switch to virtual strem.
	0x0060000A	The IP channel is offline.	Check if the IP channel is online and try again.
	0x0060000B	Live view transcoding is not supported by the device.	Use other stream type for live view.
	0x0060000C	Channel-zero is not enabled.	Enable channel-zero before starting live view of channel-zero.

Error String	Error Code	Description	Debugging Suggestion
	0x0060000D	Transcoding capability exceeded limit.	Reduce camera resolution or the number of transcoding channels.
	0x00600010	The channel does not have sub-stream.	Use main stream mode for live view.
	0x00600011	NPQ live view is not supported by the device.	Switch to other live view mode.
	0x00600012	NPQ function is disabled.	Enable NPQ function or switch to other live view mode.

### Playback Module (Error Codes Range: from 0x00700001 to 0x007fffff)

Error String	Error Code	Description	Debugging Suggestion
	0x00700001	Playback failed. Up to one channel's playback is supported.	
	0x00700002	The speed of playback displayed on video wall is not supported.	Reduce the playback speed.
	0x00700003	The transmission rate of playback stream is too high.	Reduce the transmission rate of playback stream.
	0x00700004	The encoding type of playback stream is not supported.	Provide the stream with encoding type supported by device.
	0x00700005	The container format of playback stream is not supported.	Provide the stream with container format supported by device.
	0x00700007	Exception occurred when decoding playback stream Possible reasons: displaying on video wall exception, image exception, display exception, decoding exception, image is stuck,	

Error String	Error Code	Description	Debugging Suggestion
		black screen, invalid stream type, live view is stuck, audio decoding exception, and blurred screen.	
	0x00700008	Playback video does not exit, or searching failed.	Search again or check if HDD is normal.
	0x00700009	Playback time parameter error.	Check if the time period of searched video is correct and try again.
	0x0070000A	Invalid video type.	Select the correct video type to search.
	0x0070000B	Invalid time type.	Select the correct time type to search.
	0x0070000C	Invalid event parameter.	Select the correct event parameter to search.
	0x0070000D	Invalid event type.	Select the correct event type to search.
	0x0070000E	The device does not support smart search.	Select the non smart search mode to search.
	0x0070000F	Invalid smart event type.	Select the correct smart event type to search.
	0x00700010	Invalid dynamic analysis sensitivity.	Select the correct sensitivity to search video.
	0x00700011	Reverse playback is not supported.	Select the correct playback mode.
	0x00700012	Invalid file status.	Select the correct file status to search.
	0x00700013	Invalid searching start position.	Use the correct searching start position to search.
	0x00700014	Invalid maximum number of searching.	Use the correct maximum number of searching to search.

**Capture Module (Error Codes Range: from 0x00800001 to 0x008fffff)**

Error String	Error Code	Description	Debugging Suggestion
	0x00800001	Manual capture failed.	

**Two-Way Audio Module (Error Codes Range: from 0x00900001 to 0x009fffff)**

Error String	Error Code	Description	Debugging Suggestion
startFailed	0x00900001	Starting two-way audio failed. Audio loss or driver error.	
codingFormatNot Match	0x00900002	The encoding format of the intercom is inconsistent, and the negotiation fails	Check or capture the packets on the platform, then analyze if the audio encoding formats negotiated by both sides are consistent.
dialedIsBusy	0x00900003	The intercom party is already in the intercom and can no longer respond to the intercom	Check if the intercom party is already in the intercom, if not, get the protocol message and analyze the response message.
destinationLongN umberError	0x00900004	The requested destination long number is wrong	Check or capture the packets on the platform, then analyze the long number.

**Video Storage Module (Error Codes Range: from 0x00a00001 to 0x00afffff)**

Error String	Error Code	Description	Debugging Suggestion
videoSearchFailed	0x00a00001	Searching videos failed.	No resource stored in the device.
notFindStorageM edium	0x00a00002	No storage medium found.	
videoDownloadFa iled	0x00a00003	Downloading videos failed.	

**Picture Storage Module (Error Codes Range: from 0x00b00001 to 0x00bffff)**

Error String	Error Code	Description	Debugging Suggestion
	0x00b00001	Searching pictures failed.	No picture resource.

**IO Function Model (Error Codes Range: from 0x00c00001 to 0x00cfffff)**

Error String	Error Code	Description	Debugging Suggestion
	0x00c00001	Invalid alarm input No.	
	0x00c00002	Invalid alarm output No.	

**Event Function Module (Error Codes Range: from 0x00d00001 to 0x00dffff)**

Error String	Error Code	Description	Debugging Suggestion
	0x00d00001	Incorrect event rule.	Refer to the manual for correct configuration.

**Parking Service Module (Error Codes Range: from 0x00e00001 to 0x00efffff)**

Error String	Error Code	Description	Debugging Suggestion
	0x00e00001	The vehicle with parking pass already exists.	Parking pass is created by license plate, you need to check if the parking pass for this license plate already created.
	0x00e00002	The license plate number is required.	

**General Function Module (Error Codes Range: from 0x00f00001 to 0x00ffff)**

Error String	Error Code	Description	Debugging Suggestion
noMemory	0x00f00001	Insufficient device memory (heap space allocation failed).	Check the free memory and send logs to the developer for analysis.
deviceBusy	0x00f00002	The device is busy or the device is not responding.	Send logs to the developers for analysis.

Error String	Error Code	Description	Debugging Suggestion
			For fingerprint collection, face collection, file application, and file uploading services, check if the last operation is completed.
notSupport	0x00f00003	The URL is not supported by the device.	Capture the packets, check if the applied URL exists in the PMP platform. If yes, send the URL to the developer for analysis.
methodNotAllowed	0x00f00004	HTTP method is not allowed.	Capture the packets, check the method corresponding to the URL in the PMP platform.
invalidOperation	0x00f00005	Invalid operation of API command.	
IDNotExist	0x00f00006	The ID does not exist (the URL should contain ID, but the actual URL does not contain the ID).	Capture the packets and check if the ID included in the URL is correct.
invalidID	0x00f00007	Invalid ID (the ID in the URL exceeds the capability set or the ID format is invalid).	Capture the packets and check if the ID included in the URL is correct. Get the capabilities of URL and check the ID range.
invalidIURL	0x00f00008	The content after the "?" in the URL is wrong.	Capture the packets and check if the URL is correct.
deviceAckTimeOut	0x00f00009	Device response timed out.	If the communication with the external module timed out, check if the external module is offline. When the above situation is eliminated, send logs to the developer for analysis.
badXmlFormat	0x00f0000a	XML format error	

Error String	Error Code	Description	Debugging Suggestion
badJsonFormat	0x00f0000b	JSON format error	
badURLFormat	0x00f0000c	URL format error	Get the URL and check if it is correct.
badXmlContent	0x00f0000d	XML message error: <ul style="list-style-type: none"><li>• The message contains only URL but no message body</li><li>• The required node is not configured.</li><li>• Node value exceeds the range limit (incorrect node value).</li></ul>	
badJsonContent	0x00f0000e	JSON message error: <ul style="list-style-type: none"><li>• The message contains only URL but no message body</li><li>• The required node is not configured.</li><li>• Node value exceeds the range limit (incorrect node value).</li></ul>	
messageParametersLack	0x00f0000f	The required node does not exists.	
invalidSearchConditions	0x00f00010	Invalid search condition, search again.	Check if searchID is correct.
operObjectNotExist	0x00f00011	The object does not exist (for the operations about door, alarm IO, the object is not added).	Check if door lock is connected.

**Door Control Module (Error Codes Range: from 0x01000001 to 0x010fffff)**

Error String	Error Code	Description	Debugging Suggestion
multiAuthentication Failed	0x01000001	Multi-factor authentication status operation failed.	
securityModuleOffline	0x01000002	The safety door control module is offline and fails to open the door.	Check if the safety door control is offline.

**Schedule Template Module (Error Codes Range: from 0x01100001 to 0x011fffff)**

Error String	Error Code	Description	Debugging Suggestion
planNumberConflict	0x01100001	Plan number conflict.	
timeOverlap	0x01100002	Time period conflict.	Check the message to find out if there is a time overlap of different time periods in one day.

**Person Information Module (Error Codes Range: from 0x01200001 to 0x012fffff)**

Error String	Error Code	Description	Debugging Suggestion

**Certificate Module (Error Codes Range: from 0x01300001 to 0x013fffff)**

Error String	Error Code	Description	Debugging Suggestion

**Security Function Module (Error Codes Range: from 0x01400001 to 0x014fffff)**

Error String	Error Code	Description	Debugging Suggestion
decryptFailed	0x01400001	Decryption failed, when decrypting sensitive	The import secret key should be consistent with the export.

Error String	Error Code	Description	Debugging Suggestion
		information fields or importing data files.	
certificateNotmatch	0x01400003	Certificates mismatched, SSL/TLS public and private keys need to be matched in pairs.	The public and private keys need to be generated at the same time.
notActivated	0x01400004	Device is not activated.	Activate the device by tools such as SADP before use.
hasActivated	0x01400005	Device has been activated.	
forbiddenIP	0x01400006	IP address is banned	IP address is banned when illegal login attempts exceed the upper limit.
bondMacAddressNotMatch	0x01400007	The MAC address does not match the user.	Check if the specific MAC address has linked to the user.
bondIpAddressNoMatch	0x01400008	IP address does not match the user.	Check if the specific IP address has linked to the user.
badAuthorization	0x01400009	Triggered by illegal login	Incorrect password triggered the illegal login.

### Advertising Function Module (Error Codes Range: from 0x01500001 to 0x015ffff)

Error String	Error Code	Description	Debugging Suggestion
materialDownloadFailed	0x01500001	Material download failed.	<ul style="list-style-type: none"> <li>• Check if the network connection is normal.</li> <li>• Check if the device is running normally.</li> <li>• Check the log print.</li> </ul>
materialNumberIsOver	0x01500002	The number of materials in the program list reached the upper limit.	Check if the number of materials in applied program list exceeded the limit.

